

University of Trento
University of Brescia
University of Bergamo
University of Padova
University of Trieste
University of Udine
University IUAV of Venezia

Ing. Carlos Maximiliano Giorgio Bort

ON APPLICATION OF OPTIMAL CONTROL TO INTELLIGENT MANUFACTURING

Prof. Ing. Paolo Bosetti

2013

UNIVERSITY OF TRENTO

On Application of Optimal Control to Intelligent Manufacturing

Ph. D. Head's Prof. Ing. Davide Bigoni

Final Examination 09 / 12 / 2013

Board of Examiners

Prof. Ing. Sergio Pellegrino (California Institute of Technology)

Prof. Ing. Sergio Savaresi (Politecnico di Milano)

Prof. Ing. Pasquale Russo Spena (Libera Università di Bolzano)

SUMMARY - SOMMARIO

A framework to increase the level of automation of manufacturing processes towards the paradigm of intelligent manufacturing is proposed. The case study considered is the milling of aluminum alloys through a three-axes Computer Numerical Control (CNC) machine tool. The developed controller is called Evaluation and Perception Controller (EPC), and it consists of three layers: sensor, perception, and cognitive. A set of sensors displayed in the working volume collect the information necessary to reconstruct the state of the system. In the perception layer the acquired data are processed and learned, thus maintaining updated the models of the process which drive the process optimisation in the cognitive layer.

The Optimal Control Problem (OCP) method is utilised to calculate the controls of the process that optimise a target function, defined accordingly to the specific context in which the operation is executed (namely roughing or finishing). In the general case, the objective function takes into account: productivity, quality, and costs of the process. The material removal rate along the tool path is used as index for process productivity. The quality is quantified through measurements of roughness, therefore it is improved by limiting the static and dynamic displacements (i.e. the vibrations) of the tool. The costs of the process here considered are those associated to tooling (i.e. wear of the tool), and energy absorbed by the spindle during on-air free movements.

In order to ensure the feasibility of the solution, algebraic and differential constraints are imposed on the dynamic response of drives and spindle. The OCP is then solved through an efficient optimisation library developed by the group of Mechatronic Engineering of the University of Trento.

The EPC has been designed as a portable system that can be integrated into any CNC machine, once it has been calibrated, and a dedicated communication interface with the NC has been implemented. The tests necessary for the OCP calibration, and requirements for the communication layer between EPC and CN, are described and discussed. Finally, the EPC is tested on real milling processes. The validation of its performances is done by comparing the outcomes of the process with respect to a nominal case, in which the process is set up according to guidelines given by tool manufacturer.

It worth be noted that this work represents not only a step foreword in increasing the level of automation of machining, but rather it proposes an architecture and an approach which can be generalised to several manufacturing processes.

In questo studio è proposto lo sviluppo di un controllore al fine di incrementare il livello di automazione dei sistemi manifatturieri, attraverso il paradigma della produzione intelligente. Il processo di riferimento è la fresatura delle leghe di alluminio mediante macchine utensili

a controllo numerico (CN). Il sistema sviluppato è chiamato Evaluation and Perception Controller (EPC), ed è costituito da tre livelli: sensori, percezione, e cognitivo. Sensori posizionati nel volume di lavoro collezionano informazioni sullo stato del sistema. Nel livello di percezione i dati acquisiti sono elaborati ed appresi, mantenendo così aggiornati i modelli del processo. Questi ultimi sono successivamente utilizzati nel modulo cognitivo per il calcolo dei controlli ottimi.

La tecnica del controllo ottimo (OCP) è utilizzata per individuare gli ingressi di processo che minimizzano una funzione obiettivo, definita in base al contesto specifico nel quale la lavorazione è effettuata (ovvero sgrossatura o finitura). Nella sua formulazione più generale, la funzione target include termini legati a: produttività, qualità, e costi di processo. Il tasso di asportazione del materiale lungo il percorso utensile è usato per descrivere la produttività del processo. La qualità è quantificata da misure di rugosità, ed è migliorata limitando gli spostamenti statici e dinamici (vibrazioni) dell'utensile. I costi di processo considerati sono quelli associati alla gestione degli utensili (alla loro usura), e all'energia assorbita dal mandrino durante i movimenti in aria.

Il comportamento dinamico degli assi e del mandrino è rappresentato mediante vincoli algebrici e differenziali, al fine di assicurare la controllabilità del sistema. L'OCP è quindi risolto attraverso una libreria di ottimizzazione efficiente, sviluppata dal gruppo di Ingegneria Meccatronica dell'Università degli Studi di Trento.

L'EPC è stato progettato con lo scopo di essere un sistema facilmente esportabile e integrabile in diversi controlli numerici, qualora esso sia stato calibrato e una opportuna interfaccia di comunicazione con il CN sia stata implementata.

Infine, l'EPC è testato su fresature reali. La validazione delle prestazioni del sistema qui sviluppato è svolta confrontando gli osservabili di processo con quelli ottenuti in un caso nominale, nel quale i parametri sono scelti secondo indicazioni fornite dal produttore degli utensili di taglio.

Questo lavoro di tesi rappresenta non solo un passo nell'ulteriore incremento del livello di automazione delle lavorazioni per asportazione di truciolo, ma una metodologia ed un approccio che possono essere esportati a diversi processi produttivi.

Acknowledgements

In ogni progetto la chiave per il successo è data dalla coesistenza di tre condizioni: impegno, risorse e opportunità. L'impegno è determinato dall'individuo, mentre le risorse disponibili e le opportunità dipendono dalle condizioni al contorno. Questo lavoro di tesi non sarebbe stato possibile senza la fiducia ed il supporto del mio supervisore prof. Paolo Bosetti, ma anche dei prof. Enrico Bertolazzi e prof. Francesco Biral. Li ringrazio per avermi continuamente accompagnato ed ispirato durante il mio dottorato. Ringrazio inoltre la prof.ssa Stefania Bruschi con cui ho lavorato in una prima fase di questo percorso.

Grazie al mio compagno di avventure Antonio Di Dino con il quale ci siamo continuamente confrontati, spingendoci ad affrontare gli studi da punti di vista originali e trasversali. Innumerevoli sono stati i momenti di ilarità che abbiamo condiviso assieme a Frego e Galva, e che mi hanno aiutato ad essere positivo anche nelle situazioni più difficili.

Essenziale è stato l'affetto ed il supporto di mio fratello Dino e dei miei amici di vecchia data — Marco, Daniele, Chiara, Michela ed Alice —, ma anche dei nuovi che ho conosciuto in questi ultimi anni. Ho trascorso insieme a voi momenti indimenticabili, di cui custodirò gelosamente il ricordo.

Infine, voglio sottolineare che tutto ciò che c'è di buono nella persona che sono diventato è dovuto ai sacrifici ed all'immenso amore di mia madre Diana e mio padre Carlos. L'impegno che ho messo, e metterò nell'affrontare le sfide della vita è continuamente guidato dal loro amore. A loro va tutto il merito del mio successo personale.

Los amo.

Dedication

A mi madre Diana, padre Carlos y hermano Dino que constantemente sostienen mi corazón con su amor.

(To my mother, father, and brother who constantly sustain my heart with their love.)

‘Once you have the vision and the clear thought together, what’s missing is just really good execution. And execution to me is all about the way you would climb a mountain you’ve never climbed before. If you waver along the way, if you debate, if you become uncertain about the objective, then you’re not going to make it. It’s important that you keep climbing. And it’s important that you acknowledge that you don’t have all the answers. So you will make mistakes, and you will have to back up, learn, and improve. That is a normal component of the innovative process. But you should not change your goal.’

S. Thrun (2013).

Contents

Ringraziamenti	i
1 Introduction	14
1.1 Motivation and Objectives	14
1.1.1 Sensing layer	18
1.1.2 Perception layer	19
1.1.3 Cognitive layer	21
1.2 Overview of Milling Process	23
1.3 Adaptive Controllers	27
1.3.1 Adaptive Control with Constraints	28
1.3.2 Geometry Adaptive Control	29
1.3.3 Adaptive Control with Optimisation	33
1.4 Statement of Originality	35
2 Optimal Control Problem	38
2.1 Theoretical primer	38
2.2 Formulation	39
2.3 Solution	42
2.4 OCP for milling processes	45

2.4.1	Productivity	46
2.4.2	Quality	47
2.4.3	Costs	56
2.4.4	DAE constraints	59
3	Simulacrum	61
3.1	Milling machine Alesamonti MB63	62
3.2	Numerical Control FIDIA C20	65
3.3	Sensors	65
3.3.1	Dynamometric table	67
3.3.2	Accelerometer	69
4	Evaluation and Perception Controller	70
4.1	Process simulator	74
4.1.1	Ruby Numerical Control	74
4.1.2	Cut simulator	80
4.2	OCP solver	80
4.3	Learning layer	81
4.3.1	Sensor Fusion	85
4.3.2	Validation	86
5	Communication Layers	90
5.1	Common Object Request Broker Architecture	91
5.2	Interface Ruby - CORBA	92
5.3	Characterisation and validation	95
5.3.1	Task Execution Time	96

5.3.2	Validation of the acquisitions	98
5.3.3	Latencies and sampling frequencies	98
6	Machine Identification	103
6.1	Dynamics of axes	104
6.1.1	Maximum feed rates and accelerations	104
6.1.2	Transients of feed rates and accelerations	107
6.2	Dynamics of the spindle	113
6.2.1	Spindle acceleration	113
6.2.2	Mechanical torque	114
6.3	Dynamic compliance	120
6.3.1	System identification	122
6.3.2	Regression of the dynamic compliance	126
6.3.3	Validation of the regression	127
6.3.4	Validation of the stability lobes diagrams	131
7	Applications of Augmented Reality for Safe Manufacturing	134
7.1	State of the art	135
7.2	3D scene reconstruction	136
7.3	Exploitations	140
8	Results	144
8.1	Offline optimisation	145
9	Conclusion	150
9.1	Summary of Thesis Achievements	150
9.2	Future Works	151

CONTENTS

Appendices	153
A Gaussian Processes	154
B Calculation of Stability Lobes diagram	157
C Normal Distribution Transform	164
D Identification of the cutting constants of the workpiece	167
E Technical drawings of the Alesamonti MB63	169
F Information logged by the EPC	171
Bibliography	171

List of Symbols

cs [mm²] = Cross section
 d [mm] = Depth of cut
 E_c [W] = Cutting energy
 f [m/s] = Feed rate
 F [mm/min] = Feed rate
 $\dot{f}(\zeta)$ [m/s²] = Feed acceleration
 F_c [N] = Cutting force
 f_z [mm/tooth] = Feed per tooth
 K_t [N/mm²] = Tangential cutting constant
 K_r [N/mm²] = Radial cutting constant
 l [m] = Length of the ram
 MRR [mm³/s] = Material Removal Rate
 n_t [\] = Number of flutes in the mill
 P_c [W] = Cutting power
 r_f [mm] = Fillet radius of the tool
 r_t [mm] = Radius of the tool
 S [rpm] = Spindle speed
 t [s] = Time
 T_c [Nm] = Cutting torque
 u_s [J/mm³] = Specific energy of the workpiece
 v_c [m/min] = Cutting speed

$\omega(\zeta)$ [rad/s] = Spindle speed
 $\dot{\omega}(\zeta)$ [rad/s²] = Spindle acceleration
 ζ [m²] = Curvilinear abscissa

Remaining quantities will be defined in the text.

List of Acronyms

ACC = Adaptive Control with Constraints
ACO = Adaptive Control with Optimisation
ANN = Artificial Neural Network
AR = Augmented Reality
BVP = Boundary Value Problem
CNC = Computer Numerical Control
CORBA = Common Object Request Broker Architecture
DAE = Differential Algebraic Equations
EPC = Evaluation and Perception Controller
GAC = Geometry Adaptive Control
GP = Gaussian Process
ICP = Iterated Closest Point
MPC = Model Predictive Control
MRR = Material Removal Rate
NC = Numerical Control
NLP = Non Linear Programming
NDT = Normal Distribution Transform
OCP = Optimal Control Problem
PSD = Power Spectrum Density
RNC = Ruby Numerical Control
SLAM = Simultaneous Localisation And Mapping
SLD = Stability Lobes Diagram
SSV = Spindle Speed Variation
UDP = User Datagram Protocol
VR = Virtual Reality

List of Figures

1.1	Architecture of the EPC.	17
1.2	Examples of traditional machining processes (1).	25
1.3	Damped tool.	31
1.4	Examples of weights of three terms in a generic target function contextualised on different applications.	34
2.1	Model of the cantilever beam with the cutting force applied at the free end.	47
2.2	Engagement arcs definition w.r.t. feed velocity.	51
2.3	Discretisation of the tool path (right) and computation of the target function at each quantised position increment (left).	52
3.1	Architecture of the simulacrum.	61
3.2	Configuration of the machine MB63.	63
3.3	Kinematic chain of the cutting head.	64
3.4	Scheme of the measurement system.	66
3.5	The dynamometric table.	67
4.1	Architecture of the EPC.	71
4.2	State machine of the EPC.	72

4.3	Trapezoidal (acceleration-limited) feed velocity profile. Short blocks can result in triangular profiles. Ticks on time axis represent the quantisation time t_q	75
4.4	Function $\lambda(t)$ evaluated with a sampled time of 0.01 s, $\Delta t = 1.2$ s, and maximum accelerations $A = -D = 5 \text{ m/s}^2$; vertical dashed lines limit the constant feed rate zone	78
4.5	Representation of circular interpolation.	79
4.6	Architecture of the OCP code generator and solver.	81
4.7	Validation of the learning layer for the full immersion face milling. . . .	87
4.8	Validation of the learning layer for the face milling with increasing cutting width.	88
4.9	Validation of the learning layer for the face milling with complicate tool path.	89
5.1	Architecture of the software developed for the communication between EPC and NC.	92
5.2	Communication interface and layer between EPC and NC.	93
5.3	Classes architecture of the communication layer between EPC and NC. . .	94
5.4	Histograms of the probability density distributions of TETs measured when reading the status of the machine (left), writing two parameters on the NC (centre), and performing both operations (right).	97
5.5	Comparison of the measurements of positions and velocities for the three axes, made with the EPC and with the oscilloscope internal to the user panel of the machine.	99
5.6	Feed rate profiles used to estimate the latencies measured when acquiring from the NC at $\approx 10 \text{ Hz}$ (left), and $\approx 100 \text{ Hz}$ (right).	101
6.1	Feed rates of the axes for the test performed giving as input: feed steps from 0 mm/min to 10000 mm/min (red), and rapid movements (blue). . . .	105
6.2	Accelerations of the axes for the test performed giving as input: feed steps from 0 mm/min to 10000 mm/min (red), and rapid movements (blue). .	106

6.3	Tests performed to identify the duration of the acceleration transient. .	108
6.4	Set and measured sweep profiles on the feed rate. In the test performed along Y going from the top to the bottom the feed rate is negative because positive Y are upwards.	110
6.5	Example of a deceleration test performed starting from different initial feed rates.	111
6.6	Results obtained from the deceleration tests for the X (left), Y (centre), and Z (right) axes.	112
6.7	Measured spindle speed (top), and calculated accelerations (bottom) for one of the three tests performed.	114
6.8	Measured free run mechanical torque at different positions of the ram. .	115
6.9	Measured and fitted free run torque for the test with the ram fully extended (top), and residuals of the regression (bottom).	116
6.10	Scaling factor K estimated for four acceleration/deceleration tests. . . .	118
6.11	Measured mechanical torque (top), and cut cross section (bottom). . .	119
6.12	Direction of impacts on the tool.	121
6.13	Direction of impacts on the workpiece.	121
6.14	Real part of the FRF matrix for the impacts on the tool, at different positions of the axes (see Tab. 6.2).	123
6.15	Real and imaginary part of direct FRF along X (left) and Y (right), at different positions of the ram.	124
6.16	Real part of the measured (left) and regressed (right) $FRF_{y,y}$	128
6.17	Stability lobes diagram calculated for the ram at the position -220 mm (left) and chatter frequencies (right).	129
6.18	Stability lobes diagrams calculated from measured FRFs (left), and from regressed FRFs (right).	130
6.19	Position of the accelerometer.	131
6.20	Stability lobes diagrams validated at two positions of the ram: -300 mm (top) and -270 mm (bottom). The size of the circles is proportional to the calculated CI_{ER}	133

LIST OF FIGURES

7.1	Normal Distribution Transform algorithm applied to the alignment of two 3D geometries not explicitly correspondent.	139
7.2	Conceptual architecture of the system for the collision avoidance. . . .	141
7.3	Scene reconstructed by using the Kinfu library (left) and RGB image of the working volume (right).	142
7.4	Conceptual design of an augmented interface for milling machines. . . .	143
8.1	Tool paths tested for the offline optimisation.	145
8.2	Nominal and optimised controls for the face milling.	147
8.3	Nominal and optimised controls for the peripheral milling.	148
8.4	Roughness maps for the face milling.	149
8.5	Roughness maps for the peripheral milling.	149
B.1	Two-dimensional dynamical model of milling process.	158
D.1	Cutting forces at different feed rates. The components of the force are parallel (left) to the feed, perpendicular (centre), and along the axis of the tool (right).	168
E.1	Lateral view.	169
E.2	Frontal view.	170

List of Tables

1.1	Yardstick for automation according to Amber and Amber (2).	16
3.1	Technical specifications of the MB63.	63
3.2	Technical specifications of the spindle.	63
3.3	Technical specifications of the shafts in the kinematic chain, Z is the number of teeth of the gear connected to the shaft, and D is the diameter of the pulley. The missing information can be extracted from the CAD drawings in Appendix E.	64
3.4	Technical specifications of the working table.	68
3.5	Technical specifications of the piezoelectric force links Kistler 9347C.	68
3.6	Technical specifications of the charge amplifier Kistler 5073A311.	68
3.7	Technical specifications of the accelerometer Dytran 3213M6.	69
5.1	Measured TETs for different operations, with 95% of confidence.	97
5.2	TETs required to write one parameter on the NC according to the NC manufacturer FIDIA.	98
5.3	Measured latencies.	101
6.1	Maximum velocities acceleration and deceleration of the axes.	106
6.2	Positions at which the FRF of the tool are tested, expressed in the absolute frame of reference of the machine.	122

6.3	Trained hyperparameters. $\ nlml\ $ is the module of the negative log marginal likelihood, the higher it is, the more accurate the prediction is.	127
7.1	Examples of 3D sensors available to the market.	140
8.1	Data of the utilised tool.	144
8.2	Cutting times achieved with and without the EPC.	146
D.1	Process parameters used for the cutting force constants identification. .	167
D.2	Cutting force constants for the <i>AA 6082-T6</i>	168
F.1	Codes of the machine status parameter.	171
F.2	Structure of the Status C-struct.	172
F.3	List of parameters and sub parameters listened by the Listener instance.	173

Chapter 1

Introduction

This study is focused on applying the optimal control to machining process, and it has been carried out within the Italian project Michelangelo. The main objective of this work is to increment the level of automation of manufacturing machines towards the concept of *intelligence manufacturing*.

The project has lasted three years and it has been completed on November 2013. It has involved several academic and industrial partners: University of Trento, University of Brescia, University of Bergamo, CSMT, CNR ITIA, Alesamonti S.r.l., Fidia S.p.A., Imatecno, Mandelli S.p.a., Pomini Tenova S.p.a., Scuola Superiore Sant'Anna of Pisa and Stylmeccanica.

1.1 Motivation and Objectives

The 20th Century has been characterised by the growth of economies of scale where the production costs tended to reduce with the increasing size of enterprises. As a conse-

quence, a rising number of manufacturing companies started to move to big production rates, where large manufacturing systems were exploited to fulfill the product demand of global market.

In the last 20 years, the rapid evolution of products, changing of user needs, and global competition, have forced companies to redesign manufacture chains thus taking into account not only the mere productivity, but also the quality, economy, and the flexibility of the production. These reasons have motivated an increasing number of enterprises to research and evolve their production systems towards paradigm of *intelligence manufacturing*. Increasing the level of automation in manufacturing systems generates a technological and economical evolution on a wide sectors of production chains. Optimal control can be exploited to optimise multi-objective target functions defined in order to reduce production times, costs, and increase the quality. Small companies with custom productions need high flexibility and might benefit mainly from reduced setup costs. Large production instead can achieve high productivity from automated, integrated, short-loop process control. The cooperation of multiple intelligent systems can moreover ease the definition and calibration of reliable models of the process, thus improving the goodness of the calculated optimal controls.

In this scenario, the Computer Numerical Controlled (CNC) machine tools have become popular and have played a key role in workshops, leading to a vast area of research on modelling and optimisation of machining processes. However, despite the continuous performance enhancements, since the first CNC machine tool was introduced in 1952 (3) the automation level has not been evolving as much as in other mechatronic fields, like: robotic vehicles, planes, anthropomorphic manipulators, and humanoids. In 1962 Amber and Amber defined the yardstick for automation, Tab. 1.1.

Advanced robots have perception layers to sense, reconstruct, and learn their state; they can plan online the future actions to take in order to achieve a specified goal, and can cooperate with others systems to enhance their knowledge. Such systems are typi-

Table 1.1: Yardstick for automation according to Amber and Amber (2).

Order	Human Attribute Replaced	Examples
A(0)	<i>None</i> : lever, screw,	Hand tools, manual machine
A(1)	<i>Energy</i> : muscles replaced with power	Powered machines and tools, Whitney's milling machine.
A(2)	<i>Dexterity</i> : self-feeding	Single-cycle automatics.
A(3)	<i>Diligence</i> : no feedback but repeats cycle automatically	Repeats cycle; open-loop numerical control or automatic screw; transfer lines.
A(4)	<i>Judgment</i> : positional feedback	Closed loop; numerical control; self-measuring and adjusting; CAD, CAM.
A(5)	<i>Evaluation</i> : adaptive control deductive	Computer control; model of process required for analysis; feedback from the process analysis and optimisation with data from sensors.
A(6)	<i>Learning</i> : from experience	Limited self-programming; some artificial intelligence (AI); expert systems.
A(7)	<i>Reasoning</i> : exhibits intuition; relates causes and effects	Inductive reasoning; advanced AI in control software.
A(8)	<i>Creativeness</i> : performs design work unaided	Originality creates new process programs, neural networks; fuzzy logic.
A(9)	<i>Dominance</i> : supermachine, commands others	Machine is master (Hal in 2001, A Space Odyssey).

cally called *multi agents systems* (4; 5), and in the Amber and Amber yardstick have an order of $A(6)$. The most advanced marketed CNC machines have only the judgement attribute, corresponding to an order of $A(4)$, with controllers that can compensate errors in positioning the axes, thermal distortions of the structure, and regulate the spindle speed at a nominal value. On the market adaptive controls are also available to mitigate vibrations and improve the quality of the process. Nevertheless, these systems rely only on the torque measured at the drives, and are simple feedback controls rather than proper model-driven intelligent systems. Intelligent systems are based on at least two factors: the ability to *learn* the process, and the use of a model of the process to

do *predictions*. When an agent is a machine tool, these features ensure the operation at optimal cutting conditions, which are constantly identified based on a theoretical knowledge of the process and on the current state of the machine.

The purpose of this study is to increase the level of automation of CNC machines to $A(6)$, by designing a controller whose general framework architecture could be exported to other processes than machining. The proposed architecture is called *Evaluation and Perception Controller* (EPC). It is shown in Fig. 1.1 and it closes the feedback loop on the machine through three layers: sensor, perception, and cognitive.

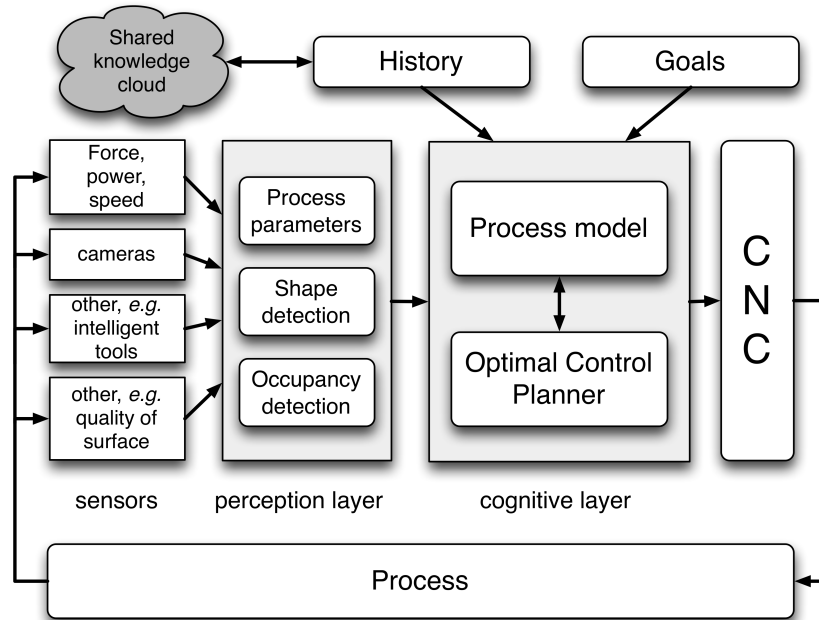


Figure 1.1: Architecture of the EPC.

1.1.1 Sensing layer

In the sensing layer the status of the machine is collected directly from the CNC, and from sensors displayed in the working volume (6). The former has the advantage that is a cheap solution, since the information on the cutting power and tool condition is obtained by measuring the motor power and current at the electric drives and spindle. These measurements however are given by the feedback controllers of the drives, therefore they tend to be affected by high noise and have usually a low sampling frequency. Additional sensors can be utilised to obtain reliable and accurate data, but due to their high costs they are usually used only in high-end and research applications. As instance, cutting force and torque can be measured through piezoelectric sensors and strain gauges placed on the tool holder. These devices can be exploited also for measuring the vibrations of the tool during the cut, however, if only such information needs to be monitored, a more suitable and cheap solution would be to use acoustic emission measuring systems (i.e. microphones), gap sensors, or accelerometers. Nevertheless, microphones should be used only when a single machine is cutting, since they might record not only the sound generated by the monitored process, but also the sound generated by others machines working in the same workshop.

Recently, thanks to the improvements of the performances of personal computers, a new class of sensors have been utilised that are based on image processing techniques. Images and videos of the working volume can be acquired through cameras, which can be classified as: video, stereo, Time of Flight (ToF), and 3D depth sensors. These devices are used to *augment* the set of information provided to CNC users, thus offering tools to monitor, supervise, and maintain remote processes, as well as to test complex collision-free tool paths and to ease the training of novice workers (7).

1.1.2 Perception layer

The perception layer fuses the sensed data to reconstruct the state of the system, which is defined by the instantaneous process status, cut cross section, and tool condition. Force, acceleration, gap, and acoustic signals can be analysed in the time as well as frequency domains (8). Measurements are usually studied through the Fourier transform method power spectra and Hilbert-Huang Transform (9), but the best state estimation is achieved when a model-based filter is used.

The most famous state estimators are *Kalman Filters* (KF) (10) and *Particle Filters* (PF) (11). These have been utilised both individually and together in numerous applications, such as: navigation of unmanned aerial and ground vehicles (12; 13; 14; 15), simultaneously localisation and mapping of robots (16; 17; 18), parameters estimation (19; 20; 21; 22), economic forecasting (23), wireless LAN tracking (24; 25), and vision-based tracking (26; 27; 28).

The state space of KF and PF is continuous, but usually these filters are considered complementary. KF is an efficient algorithm since it scales quadratically with the dimension of the state variables, and it is confined to estimate unimodal distributions. The tuning of KF however might be difficult since it requires to estimate both the measurement and model uncertainty. PF on contrary describes multimodal distributions, is easier to implement, but its efficiency might be highly affected when the dimension of the state is greater than four.

Images acquired from cameras can provide numerous information about the ongoing machining process, with a modest economic investment. High resolution cameras and efficient *Graphics Processor Units* (GPU) —which have been developed for gaming consoles and PCs— are nowadays available on the market together with open-source

state-of-the-art image processing libraries. *OpenCV* (Open Source Computer Vision) ¹ is a computer vision and machine learning software library with more than 2500 optimised algorithms that can be used to a wide variety of applications, such as: detect and recognise faces, identify objects, classify human actions in videos, image stitching, follow eye movements, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, recognise scenery and establish markers to overlay it with augmented reality. To *augment reality* it means to extend the set of information provided to the CNC user, by merging the data recorded by the NC, sensors, and video.

Collision avoidance

One of the main challenges in designing a milling or turning manufacturing process is to obtain a safe tool path, free from unexpected collisions between the tool and the workpiece. Safe tool paths are crucial, not only to avoid expenses damages of the milling machines, but also when expensive materials have to be cut. It is possible to classify the safety systems as passive or active, likewise it is done with vehicle safety systems.

Passive systems act once the collision has occurred, and they reduce the entity of the damages. These systems tend to be used in unstructured environments, but they might be not effective when collisions between expensive tools and workpieces must be avoided a priori.

Active systems instead can predict and avoid dangerous situations, therefore they are particularly suitable for machining processes, where the high inertia of the drive is involved during the cut. The prediction will be as much good as much precise is the knowledge of the supervised system. It is not trivial however to have a robust model of the working volume due to the complexity of the kinematics of the 5-axis machines, and

¹<http://opencv.org/>

shape of the workpiece. Nowadays, two strategies are followed to address this problem: the g-code can be tested in free air directly on the machine, or the tool path can be designed and tested through computer aided manufacturing (CAM) softwares (29; 30). The former approach tends to be time consuming (thus reducing the productivity), and relies on the ability of the worker man to correlate the in-air process with the real one. The robustness of CAM softwares used for collision check depends on the skills of designers to take into account several factors, such as machine kinematics, geometry of raw workpiece and tool, and accuracy of theirs CAD models. Usually, the CAD model of the milling machine is provided by the machine manufacturer, but it not so for the tool and tool holder. Raw workpieces can have complex patterns, especially if they have been generated with forging or casting processes.

Several issues must be addressed when applying image processing techniques to augment the reality of milling processes: the presence of high frequency vibrations, debris, chips, coolant liquids inside the working volume, and reflecting surfaces of the tool and workpiece. A description of the feasibility study carried out on this topic is given in Chapter 7.

1.1.3 Cognitive layer

In the cognitive layer the past history, present state and updated models of the process are used to close the loop and calculate the optimal process parameters. For small mobile robots the loop closure is done through the optimisation of simple linear models, with techniques called *Model predictive Control* (MPC) (31; 32; 33; 34), in which a reference signal is followed. As a main advantage the MPC requires little computer resources to be run online, therefore it is particularly suitable to be implemented in small controllers. However, the use of linear models represent a strict simplification for

some applications, such as machining processes. When compactness and lightness are not mandatory requirements, a personal computer can be utilised to perform non linear model-based optimal control. Example of these systems are large moving robots such as vehicles or long range drones, as well as manufacturing systems where the model of the process is formulated through differential equations. The optimal control technique allows to maximise a set of performances, hence contextualising the controller on a specified application. In robotic vehicles the target function can be defined in order to reach a goal by optimising the path and controls, while minimising the travelling time or the energy consumption (35; 36; 37).

When improving machining systems, multiple objectives can be pursued. Mechanical workshops are interested mainly in: quality, productivity, and cost reduction. Constraints on robotic systems can be also taken into account to ensure the feasibility of the optimal solution. The equations that define the target function, model of the system, and constraints form a system of *Differential Algebraic Equations* (DAE), which are numerically integrated and minimised. These operations have a high computational cost that is addressed, even in real-time, by modern personal computers. In-line optimal control can be carried out to enhance the behaviour of robots while compensating external disturbances. For milling processes, online control allows to adapt the system when the workpiece material has non homogeneous properties, or when the tool wears. The cognitive layers aims thus at collecting the knowledge of skilled workers, easing the calibration of process parameters, and eliminating time-consuming manual design of optimised part programs.

The effectiveness and reliability of intelligent systems depend on the level of knowledge programmed into the cognitive layer. Artificial intelligence techniques are used to learn and adapt models of the process from past records and, eventually, from a shared knowledge. Such approach is called *machine learning*, and the generated models can be

used for classify and regress the sensed data. The former is performed on discrete finite models, and it is done to subdivide the observed quantity into groups. The latter on contrary is performed on continuous models, and its objective is to formulate continuous functions. In any case, models can be generated through supervised, unsupervised, and reinforcement learning. When the training is supervised, labelled data are used to train and validate the model. In unsupervised training the machine finds itself the most significant patterns and features of outputs of the process that must be taken into account to describe the system. Finally, in reinforcement learning the actions done by the machine are used to explore the set of features of the environment.

The training process is, generally speaking, time consuming and requires a large amount of data. The information can be collected, inferred, and learned from a local set of sensors or from a pervasive network of systems deployed across distributed manufacturing plants. If cloud computing is used to offer *Data as a Service* (DaaS), then the resources and efforts needed to train large intelligence system can be significantly reduced. As suggested by Wang and Xu (38) as well as Wu *et al.* (39), information about process data, tools and machine performances, optimisation strategies, experience, training data, and generally speaking all the mutable information that each controller builds up with time, would possibly be moved to a cloud DaaS, thus expanding the set of features at the core of the learning process.

1.2 Overview of Milling Process

In machining processes a rigid tool is used to cut a workpiece that can be a metal, polymer, or ceramic (1). The process is flexible, can be easily exploited in dynamic production chains, and combined with other non-net shape processes such as casting, forging, and sheet metal forming to obtain complicate parts. Machining of polymers is

usually disadvantaged by low productivity, and the cut of ceramics is inefficient due to high tooling costs. Traditional machining processes can be subdivided into four classes, based on the relative movement between the tool and the workpiece: turning, milling, drilling, and grinding.

In turning axial-symmetrical geometries are generated. The workpiece spins and the tool is feed both radially and along the rotational axis. In grinding an abrasive tool is used to remove small fractions of the workpiece and to achieve high-quality surfaces. In drilling a cylindrical or conical tool spins around its axis and is fed along this direction to generate or enlarge holes. Finally, milling processes are those in which a tool, that can have a complex geometry, is spin and moved along a path on the workpiece thus generating complex patterns, Fig. 1.2. The norm of the linear velocity of the mill travelling across the workpiece is called *feed rate*.

It is not trivial to design and control machining processes and predict their outputs, due to several sources of unknowns, uncertainties, and multi-physical phenomena involved:

- The thermo-mechanical properties of workpiece and tool might be unknown.
- Workpiece and tool undergo to high stresses, strains, strain rates and temperatures. In such extreme conditions it is difficult to describe with a good approximation the real behaviour of the materials.
- Cutting forces can change periodically but also instantaneously.
- Friction forces at the tool workpiece interface are high and might saturate, therefore simple linear models cannot describe them with sufficient accuracy.
- Kinematic of the cut can be complicate since the tool and the workpiece have in general complex geometries.

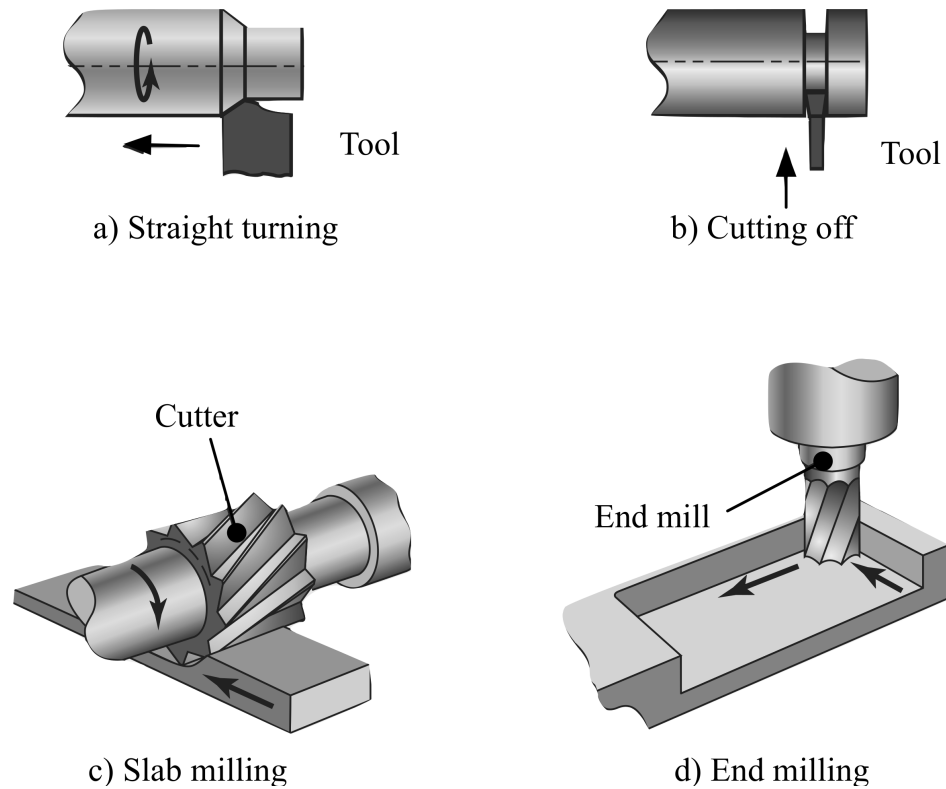


Figure 1.2: Examples of traditional machining processes (1).

- The dynamic properties of the machine, such as the stiffness and damping factors, may be unknown or difficult to characterise.
- The lubricant fluid can form an unstable film at the tool-workpiece interface.

Inputs of the process taken in into account during the setup can be distinguished into constrained and degree of freedoms. The formers are fixed by the context and specific application, and are related to: workpiece and tool materials, their geometries, the lubricant fluid to be used, the characteristics of the machine, and the tool path (i.e. desired geometry). The degree of freedoms are the rotational spindle speed S and the *Material Removal Rate* (MRR), which is correlated to the feed rate F , to the cutting

depth d , and to the cutting width w :

$$MRR = F d w \quad (1.1)$$

From the product between the MRR and the specific energy u_s of the cut material, it is possible to calculate the cutting power P_c and cutting torque T_c :

$$P_c = MRR u_s \quad (1.2)$$

$$T_c = \frac{T_c}{S} \quad (1.3)$$

F and S are the controls that must be carefully set to achieve an optimal process. These parameters must be chosen in order to be robust with respect to deviations of the cut from nominal conditions. Unpredictable inline variations can be related to:

- Varying cutting depths and widths.
- Tool wear.
- Non-homogeneous material properties.
- Irregularities of the surface of the workpiece.
- Geometrical accuracy and tolerances of the blanks.
- Chip accumulation.
- Thermal properties of the lubricant fluid can vary its behaviour.

Usually, process planners choose the cutting parameters accordingly to tabulated values reported in the catalogues of tool manufacturers. Nevertheless, it must be considered that suggested parameters tend to be conservative and far from optimal. F and S are

chosen such as they take into account and mitigate all the uncertainties and variability associated to the process, through a decision process based on the experience of most skilled technicians rather than on theoretical and methodological considerations.

1.3 Adaptive Controllers

Adaptive controllers (AC) are devices that regulate process parameters to achieve a certain performance. In the case of milling process, the controls that AC regulate are the spindle speed and feed rate. A recent review of AC system was given by Stavropoulos *et al.* (40). Ulsoy and Koren (41) identified three types of AC:

1. *Adaptive Control with Constraints (ACC)*: these systems are typically aimed at maximising the cutting force within the limits imposed by the maximum stress that the tool can tolerate without breakage. Such systems are mainly used in roughing operations in which a low cutting time is sought, without considering the quality of the cut. These conditions might affect sensibly the process productivity since a high Material Removal Rate (MRR) is associated to high cutting forces, and according to Kurada and Bradley (42) the amount of downtime caused by tool breakage is around 20%.
2. *Geometry Adaptive Control (GAC)*: these controls are used in finishing cuts, where the main objective is to achieve a high-quality surface, which can be described by a high geometrical accuracy, good surface integrity, as well as low roughness.
3. *Adaptive Control with Optimisation (ACO)*: the controller developed in this study belongs to this category. The process parameters are computed and regulated

in order to optimise a certain index or performance, such as: reduction of vibration, increasing of productivity, improvement of surface quality and tool life. Constraints on the machine dynamics can be taken into account to ensure the controllability of the system.

1.3.1 Adaptive Control with Constraints

Many examples of ACC systems are available in literature. Among the most recent studies, Budak and Kops (43) developed a system for maximise the cutting force by overriding the feed-rate. The same approach, but applied to end-milling was followed by Zuperl *et al.* (44). The authors maximised the feed-rate by utilising an adaptive neuro-fuzzy inference system. Tansel *et al.* (45) and Aykut *et al.* (46) utilised a Genetically Optimised Neural Network System (GONNS). Artificial Neural Networks (ANN) were trained to represent the relationship between surface roughness, cutting forces, and process inputs, and a Genetic Algorithm maximised the MRR while keeping the cutting forces below a pre-determined threshold. Such approaches however, required a time consuming training of supervised learning algorithms used to model the relation between feed rate and cutting force. This issue did the system poorly suitable to be applied in real workshops where several materials were cut. Sugita *et al.* (47) realised a optimal control for bone machining where the feed rate was maximised by using two different policies along the tool path: an "air-cutting mode" and a "force control mode". The latter was used only along the parts of the tool path in which the mill was in contact with the workpiece. The drawback of this method, was that it required the use of an expensive external sensor to measure the cutting force. This problem was solved by Kim and Kim (48), as well as Wang *et al.* (49) which estimated the three components of the cutting force from the current absorbed by feed-drive servo motors.

1.3.2 Geometry Adaptive Control

Studies carried out in order to optimise the quality of the cut are aimed at reducing the vibrations and the thermal deformations of the machine during the process.

Hao *et al.* (50) treated thermal errors as the superimposition of a series of error modes obtained from five measurements of temperatures of a turning centre. Neural network combined with genetic algorithm were used to predict and compensate the thermal deformation of the structure. Model-based prediction of structural deformations however can be limited from the geometrical and physical complexity of the model. Yang and Ni (51) updated in real-time the coefficients of the thermal-error model through a Kalman Filter. A different approach was followed by Bosetti and Bruschi (52) whom developed a new measurement system to directly provide real-time information about the deformation of the structure. The system was called reticular displacement measurement system (RDMS) and it was based on an array of strain sensors acquired in real-time. The main advantage of this system was its independency from expensive, sensitive, and error-prone models.

The second source of geometrical errors, especially for what concerns the surface roughness, are the vibrations of the machine. These can be associated to the compliance of the workpiece, tool, and even of the axes of the machine. Vibrations are due to the impacts of the tool entering the workpiece, and can be classified into forced, or chatter. The former are due to the excitation of the natural frequencies of the tool-workpiece system by the discontinuous cutting forces. In this case the dynamics of the system is stable and tends to be damped. Chatter instead can be generated by two mechanisms: regeneration and mode coupling. Regenerative vibrations are due to a phase shift between vibration waves left on the inner and outer surfaces of the cut chip. Mode coupling happens when the tool traces an elliptic path along the workpiece, thus

varying periodically the depth of cut. In both cases, as the cut proceeds the dynamics of the system becomes unstable, increases progressively the chip thickness, and consequently the magnitude of the cutting forces. Such unstable condition reduces the quality of the cut, as well as geometrical accuracy of the generated surfaces, but also might be dangerous for the tool, spindle, and machine structure. Skilled CNC users set the spindle speed with an heuristic method, that is by changing the spindle override until they don't hear anymore the characteristic sound of chattering process. The value of the stable spindle speed however depends not only from the dynamical behaviour of tool, workpiece, and machine structure, but also from the entity of cutting forces which might change at every position of the tool along the tool path. One of the main contributions to the problem of self-excited vibrations was given by Altintas (53) who developed a model to determine the combination of spindle speed and depth of cut that delimited stable cutting conditions from unstable. Numerous chatter mitigation strategies were studied, which can be distinguished between passive and active (54).

Passive chatter suppression systems

Passive chatter suppression strategies are based on modifying certain machine tool elements to enhance the stability of the system. These approaches consist in changing the dynamical behaviour of the machine tool, the cutting tool, and tool holder (55). When the tool is the weakest element in the system, tuneable vibration absorbers (56; 57), impact dampers (58), or viscoelastic dampers (59) can be used. On the market there are available tool holders, in which a mass internal is used as a Frahm damper (60), Fig. 1.3. Tools manufacturers sell also non-standard mills with variable pitch and helix angles (61; 62), which increase the stability limit by disrupting the regenerative effect.

Cau *et al.* (63) used numerical simulations (represented by FE models) to design a



Figure 1.3: Damped tool.

milling machine with a-priori known dynamic performances, which were calibrated in order to avoid as much as possible regenerative vibrations in a specific range of process parameters. Such approach can be utilised when a system was designed *ex novo*, while chatter often have to be addressed on already existing machines.

Active chatter suppression systems

Active chatter suppression systems can monitor the dynamic state of the machine tool, identify unstable conditions, and act on the system to improve its performances. One of the first implemented active chatter suppression systems was developed by Dohner *et al.* (64). The authors identified spindle speeds that ensured a stable process by calculating the lobes stability diagram of the cutting system. The spindle was sensed through strain gages at tool root, and it was damped through a piezo-electric actua-

tor. Another example of active piezo-electric actuators utilised to reduce the dynamic displacement between the tool and workpiece was given by Abele *et al.* (65). Moradi *et al.* (66) proposed a H_∞ control algorithm to control the chatter under tool wear and parameter uncertainties. In a more recent study, Moradi *et al.* (67) exploited the a *Linear Quadratic Regulator* (LQR) to suppress self-regenerative vibrations. This technique allowed the authors to use a non linear model of the process in which forces and chip thickness were related by a cubic relationship. The controls were the counterbalance forces exerted by external actuators, and the target function was defined in order to limit the chatter while minimising the efforts of the controls. Model-based control techniques were exploited to identify the process parameters that allowed stable cuts. Potočník and Grabec (68) used ANN to model the process and Nonlinear Model predictive Control (NMPC) to control the cut with stable parameters.

Active systems found in literature require an external-piezo electric actuator to reduce the vibrations. To accommodate these devices on the spindle might in general require expensive mechanical modification and re-designs. A common strategy nowadays followed to mitigate the chatter is called *Spindle Speed Variation* (SSV) (69; 70; 71). It consists in continuously changing the spindle speed with a sinusoidal patten around the mean speed to disturb the regenerative mechanism. The main advantage of this technique is that it does not require expensive tool holders and it can be implemented in modern CNCs. However, the calibration of this methodology can be difficult. Hajikolaie *et al.* (72) used a genetic algorithm to set the amplitude of the speed modulations such that the input energy to the process was minimised.

In the study here presented, chatter and forced vibrations is addressed with a different approach. It is known indeed that the stability of vibrations depends on the depth of the cut and on the spindle speed. The former is usually a fixed parameter of the process, the latter instead can be calibrated. The idea is —for a given tool engagement—

to include into the multi-objective optimal control the information about the stability lobes diagram which drives the EPC in choosing stable spindle speed, as it was proposed by Bosetti *et al.* (73). The lobes stability diagrams are calculated by considering the dynamic compliance of the machine structure in the current axis configuration, and the cross section predicted by the geometrical engine simulator, as it will be explained in chapter 2.4.2.

1.3.3 Adaptive Control with Optimisation

ACOs merge the feature of ACCs and GACs, and they fall in the category of intelligent systems. In the last 20 years there has been a massive effort from researches in outlining and designing the architecture of intelligent systems applied to manufacturing. Mekid *et al.* (74) defined the requirements ACO manufacturing systems should satisfy:

1. Integration: to perform conventional and non conventional processes.
2. Bi-directional data flow: to exchange information between the NC and drives on a single machine, but also through the workshop.
3. Process control loop: to in-line monitor and control tools, parts, and processes.
4. predictive maintenance: as a basis for self-reliant machine operation.
5. Autonomous optimisation: that is the development of a self-configuring and self-optimising control system, based on a knowledge of the process constantly updated.

These systems must be reconfigurable and dynamically fuse the human and machine intelligence, as well as manufacturing knowledge and state-of-the-art models. Interconnected subsystems and functions can be used to collect information about: geometries,

status of the process, tool management, process planning, and control. These data can then be exploited to optimise a multi-objective target function, which is defined by a weighted sum of different terms taking into account process productivity, quality, and cost. The weights of the target function can be tuned accordingly to the specific manufacturing context. As instance, in roughing operations the productivity and the cost of the process will be multiplied by higher coefficients (Fig. 1.4-b). On the contrary, when finishing the workpiece the term of quality will be dominant (Fig. 1.4-c).

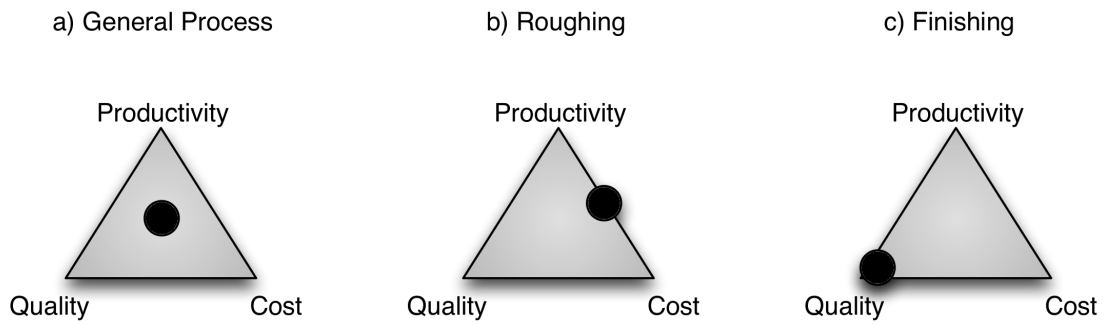


Figure 1.4: Examples of weights of three terms in a generic target function contextualised on different applications.

An early ACO systems was developed by Amitay *et al.* (75). The authors optimised the MRR in grinding, subjected to constraints on workpiece burn and surface finish. The same targets were sought by Tuming and Park (76) who designed a controller to maintain a desired peak cutting force under uncertainty of model parameters. The controller was designed through conventional and decoupled estimation methods. Kruger *et al.* (77) proposed a generic architecture of a rational agent system, integrating heterogeneous intelligent algorithms to self-learn and regulate a drilling process. ANN modelled the on-line measured data and genetic algorithms were used to find the optimal trade off between technical and economic factors. A more completed target function was formulated by Zuperl *et al.* (78), by including cutting costs and production rate (depending on MRR and tool life), while taking into account the maximum

cutting power, cutting torque, surface quality, workpiece deflection, and chip build-up edge. The same performance indexes were considered by Ridwan and Xu (79) who implemented a STEP-NC ISO 14649 (80) enabled Machine Condition Monitoring system, which was composed by three layers. The first module performed a preliminary feed rate optimisation to determine the machining parameters which optimise the cutting time and quality. The former was minimised in roughing cuts, the latter was maximised in finishing operations. A process simulator was developed to estimate the cutting power, and a fuzzy logic processed imprecise data. Online, a second subsystem carried out an adaptive feed-rate optimisation in order to keep the spindle load constant and close to the simulated value. The third module was a knowledge-based evaluation system, that was responsible of generating accurate, informative and updated process data. The authors however did not control the spindle speed, a parameter whose tuning is critical when self-regenerative vibrations need to be avoided.

1.4 Statement of Originality

Even if the devices developed in literature represent a remarkable effort in enhancing the automation level of CNC machines, they tend to be little flexible, and not practical to be used in real workshops. As instance, the models of the process rely on black-box algorithms (such as ANN) which are accurate only for a specific set of workpiece and tool materials. For every tool-workpiece combination, time consuming model trainings are required. Moreover, information about the geometry of the cut, which is necessary to calculate the cutting force, is missing. In those studies, process optimisation is typically performed by exploiting evolutionary algorithms that are suitable for non differentiable target functions and models, but require several iterations to achieve the optimal solution, and are not deterministic. It means that the convergence criteria

must be carefully defined, otherwise even when starting from the same initial guess, the algorithm will reach different optimal points. Finally, the dynamics of the system is not considered. The controls are computed without considering how the machine will react during transients, which might occur in complex tool paths.

The aim of this study is to tackle limitations of the work done up to nowadays, by developing an intelligent manufacturing system, capable to simulate, optimise, sense, and learn the milling process.

The description of the process is based on theoretical models which are enough accurate to capture the main phenomena involved during machining operations. The utilised models are well known in literature, therefore their coefficients are tabulated in manuals, and can be calibrated through simple and short cutting tests. A process simulator is integrated into the developed system to pre-calculate the instantaneous cross section and cutting force at each position of the tool. The simulator is a geometrical kernel that is commonly used in *Computer-Aided Manufacturing* (CAM) softwares. The cut cross section is then interpolated along the curvilinear abscissa with a continuous and differentiable function.

Thanks to the proposed approach it is possible to generate a continuous and differentiable target function that is minimised through efficient optimisation algorithms. Machine performances and dynamic characteristics are taken into account through a system of *Algebraic and Differential Equations* (DAE).

The set of target function and DAE constraints define the *Optimal Control Problem* (OCP). It should be mentioned that is not trivial to solve the OCP, especially when the optimal solution needs to be calculated in *real time* (i.e. at 10 Hz). Nevertheless this problem has been already addressed in robotics, and especially in autonomous vehicles navigation (37; 36; 81) where the dynamic behaviour is even more complicated and non-linear than that of machine tools.

This thesis summarises the work done in implementing the OCP for machining of metals. The intelligent system here developed is called *Evaluation and Perception Controller* (EPC). Target function and DAE constraints are defined based on models of the process that are calibrated through dedicated experimental tests. The system includes a learning layer to autonomously update the coefficients of the models based on its experience. All the test carried out to characterise the machine tool are described as a guide for future utilisations of the system. The EPC is also tested on real processes to validate its performances. Process productivity, quality, and costs are sensibly improved, thus emphasising the benefits given by increasing the automation level of machine tools.

Chapter 2

Optimal Control Problem

In this chapter is proposed a brief introduction to the *Optimal Control Problem* (OCP). It is described the approach followed to minimise a non linear target function subjected to constraints, which are expressed as a set of Differential and Algebraic Equations (DAE). Further insights can be found in (82).

It is moreover derived the formulation of the OCP applied to the milling process. Particular emphasis is given to the definition of the target function, starting from technological requirements of modern machining workshops.

2.1 Theoretical primer

The OCP is aimed at calculating the sequence of inputs that allows a dynamical system to be controlled in an *optimal* way. The target function can be defined to optimise an instantaneous performance, but also to maximise the response of the system in a future time span: the *horizon*. The feasibility of the optimal controls is ensured by a series of path, punctual and integral constraints, which can be applied to the state

variables and to the controls. Once the optimal solution has been calculated, the state of the machine is updated, the analysed horizon is shifted ahead in the future, and the OCP is solved again. By continuously updating the instantaneous value of the state variables it is possible to compensate unpredicted variations of the state of the system. This technique is called *receding horizon*. As a result, the OCP returns not only the history of the controls, but also the whole evolution of the system.

Target function and constraints are generated from a model of the process. To a more accurate description of the system corresponds a solution that is closer to the true optimum. However, complicate models can be computationally expensive and impossible to solve in real-time. The model should then be simple, but enough representative in order to capture the main phenomena at which the system is subjected.

The OCP has been investigated in a wide number of robotic systems. The main contribution of OCP has been in the field of autonomous vehicles, where it has been applied to numerous problems, such as: obstacle avoidance (83), computation of optimal path (37), time-optimal and minimum-effort control (84).

2.2 Formulation

Four are the elements that must be defined to formulate the OCP: the model of the system, the controls, the target function, and a set of constraints. The model of the system can be non linear, but it should be controllable across the whole control space. A non controllable condition might happen when it is desired to act on a system but the controls are not enough powerful, as instance when it is desired to push a heavy mass but there is not enough force available. Generally speaking, the controllability of a system might depend also on the initial conditions. In this case the system can be regulated by imposing soft constraints, that is by iteratively changing

the initial conditions. The target function can be expressed in order to optimise the initial or final condition of the system, as well as its evolution during a future time window. The former is called problem of *Mayer*, and the latter problem of *Lagrange*. A linear combination of Mayer and Lagrange target function form the problem of *Bolza*. Assuming that $x(t)$ is the state of the system and $u(t)$ is the control, the target function J can be defined as:

1. Mayer:

$$J = \phi(x(t_f)) \quad (2.1)$$

where $\phi(x(t_f))$ is the *terminal cost function*, that is independent from the controls since they are considered piecewise constant. Common definitions of the terminal cost are those used to minimise the error in reaching a desired final state, the time required to achieve a desired condition, or the fuel consumption:

$$\phi(x(t_f)) = x^T(t) P x(t) \big|_{t=t_f} \quad \text{Error in final state} \quad (2.2)$$

$$\phi(x(t_f)) = |t_{final} - t_{initial}| \quad \text{Minimum time problem} \quad (2.3)$$

$$\phi(x(t_f)) = |m_{initial} - m_{final}| \quad \text{Minimum fuel consumption} \quad (2.4)$$

where P is a square matrix of real values, and m is the mass of the system.

2. Lagrange:

$$J = \int_{t_0}^{t_f} L(x(t), u(t)) dt \quad (2.5)$$

$L(x(t), u(t))$ is the *Lagrangian function*. By integrating the Lagrangian function ahead in the future, it is possible to optimise the expected evolution of the system. $L(x(t), u(t))$ can be formulated as instance to minimise the time (i.e. it becomes equal to Eq. 2.3), the quadratic trade between state and controls, or the fuel

usage spent across the path:

$$L(x(t), u(t)) = 1 \quad \text{Minimum time problem} \quad (2.6)$$

$$L(x(t), u(t)) = x^T(t) Q x(t) + u^T(t) R u(t) \quad \text{Trade state controls} \quad (2.7)$$

$$L(x(t), u(t)) = \dot{m}(t) \quad \text{Minimum fuel usage} \quad (2.8)$$

where m is the mass of the system.

3. Bolza:

$$J = \phi(x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t)) dt \quad (2.9)$$

Constraints are applied to the minimisation of the target function, thus ensuring the reachability of a desired state and controllability of the system. The maximum value of the state variables and controls are limited through algebraic equations and inequalities. Ordinary Differential Equations (ODE) define how the system and the controls evolve in time. Boundary conditions are specified if the initial state of the system is known, and if it is desired a particular final configuration.

In this study, the independent variable is assumed to be the curvilinear abscissa ζ representing the position of the mill in the toolpath. This change of coordinate allows to univocally define the position of the tool at each optimisation instant, even for different feed rates. The complete formulation of the OCP is then:

$$\left\{ \begin{array}{ll} \min_{u(\zeta), x(\zeta)} \int_{\zeta_a}^{\zeta_b} J(x(\zeta), u(\zeta)) d\zeta & \\ s.t. & \\ \phi(\dot{x}(\zeta), x(\zeta), u(\zeta)) = 0 & \text{ODE constraints} \\ g(x(\zeta), u(\zeta)) \leq 1 & \text{One sided constraints} \\ -1 \leq h(x(\zeta), u(\zeta)) \leq 1 & \text{Two sided constraints} \\ b(x(\zeta)) = 0 & \text{Initial constraints} \\ e(x(\zeta)) = 0 & \text{Final constraints} \end{array} \right. \quad (2.10)$$

2.3 Solution

The OCP can be solved through different strategies, which might differ on how the time is discretised, which numerical scheme is adopted, and how constraints are handled. The most common methods are classified as direct and indirect. The former reformulates the OCP as a non linear programming problem (NLP) that is solved with a sequential quadratic programming (SQP). In indirects methods instead the OCP is treated as a Boundary Value Problem (BVP) through the variational calculus. Usually this approach is more complex to implement but it is also more robust and efficient.

In this study it is utilised the optimisation library *XOptima* developed by the group of Mechatronics of the University of Trento. This library exploits the symbolic engine in the software Maple v16.0 to formulate the BVP and to solve in real-time the OCP through an efficient indirect method. An comprehensive description of the algorithms implemented in *XOptima* is given by Bertolazzi *et al.* (37), and here it is briefly presented.

The inequalities constraints are eliminated with penalty formulation, and the minimisation of the target function is obtained through an affine invariant quasi-Newton method based on Broyden update formula with globalisation strategy.

Barrier functions are defined to eliminate inequality constraints in Eq. 2.10:

$$p_{1,\infty}(x) = \begin{cases} 0, & x \leq 0 \\ \infty, & x > 0 \end{cases} \quad \text{and} \quad p_{2,\infty}(x) = \begin{cases} 0, & -1 \leq x \leq 1 \\ \infty, & \text{otherwise} \end{cases} \quad (2.11)$$

The OCP then becomes:

$$\begin{cases} \min_{u(\zeta), x(\zeta)} \int_{\zeta_a}^{\zeta_b} J(x(\zeta), u(\zeta)) + \sum_{k \in \mathcal{J}_1} p_{1,\infty}(h(x(\zeta), u(\zeta))) + \sum_{k \in \mathcal{J}_2} p_{2,\infty}(g(x(\zeta), u(\zeta))) d\zeta \\ \quad s.t. \\ \phi(\dot{x}(\zeta), x(\zeta), u(\zeta)) = 0 & \text{ODE constraints} \\ b(x(\zeta)) = 0 & \text{Initial constraints} \\ e(x(\zeta)) = 0 & \text{Final constraints} \end{cases} \quad (2.12)$$

where \mathcal{J}_1 and \mathcal{J}_2 are two different sets of indexes. Lagrange's multipliers are used to eliminate differential-algebraic equality constraints and search for unconstrained minima of Eq 2.12. For sake of notation, dependency of multipliers, state, and controls variable from ζ is omitted. The sum of the penalty functions is condensed into a single term called *augmented lagrangian term* $J_p(x, u; p)$:

$$\begin{aligned} \min_{u, \mu, x, \lambda, \nu} \mathcal{F}(u, \mu, x, \lambda, \nu; p) &= \int_{\zeta_a}^{\zeta_b} \lambda \phi(\dot{x}, x, u) + J(x, u, p) + J_p(x, u, p) d\zeta \\ &\quad + \mu b(x(\zeta_a), p(\zeta_a)) + \nu e(x(\zeta_b), p(\zeta_b)) \end{aligned} \quad (2.13)$$

$J_p(x, u; p)$ is the most important function since it describes the whole OCP. The BVP is then given by the first variation of $\mathcal{F}(u, \mu, x, \lambda, \nu; p)$ and, by setting $y = (x, \lambda)^T$, it is possible to formulate it as:

$$a(\dot{y}(\zeta), y(\zeta); p(\zeta)) - \frac{d}{d\zeta} b(y(\zeta); p(\zeta)) = 0 \quad s \in (\beta_k, \beta_{k+1}) \quad (2.14a)$$

$$\lim_{l \rightarrow \beta_k^-} \lim_{\rho \rightarrow \beta_k^+} j(y(l), y(\rho); p(l), p(\rho)) = 0 \quad k = 1, 2, \dots, d \quad (2.14b)$$

$$h(y(a), u; p(a)) = 0 \quad (2.14c)$$

$$g(y(b), \nu; p(b)) = 0 \quad (2.14d)$$

$$(2.14e)$$

Eq. 2.14a is given from the first variation of Eq. 2.13. Eq. 2.14b is used to regularise internal jumps given when $p(\zeta)$ is discontinuous at β s points. For ease the solution of the problem, is set $\beta_0 = \zeta_a$ and $\beta_{d+1} = \zeta_b$. Eqs. 2.14c and 2.14d are the boundary conditions of the degrees of freedom and Lagrange's multipliers. These variables can be collected into the vector Z .

The Jacobian of the BVP is approximated through a finite difference scheme, and a highly non-linear system $\Phi(Z; P)$ is obtained. The solution of $\Phi(Z; P)$ is calculated by minimising $\|\Phi(Z; P)\|^2$. In XOptima the minimisation is carried out through a damped Newton algorithm with line search. This algorithm requires the computation of the Jacobian matrix $J_l = \partial_Z \Phi_l$ with $\Phi_l = \Phi(Z_l; P)$, and its scheme can be summarised as:

$$\begin{aligned} \text{Compute search direction } d_l \text{ by solving : } J_l d_l &= -\Phi_l \\ \text{Update } Z : Z_{l+1} &= Z_l + \alpha_l d_l \end{aligned} \quad (2.15)$$

The robustness of the procedure is ensured by the damping factor α_l , which satisfies the Armijo conditions (85) and makes the algorithm affine and invariant. It should be noted that the main fraction of computational resources in indirect methods is required

to calculate the Jacobian matrix and its factorisation. To speed up the optimisation, exact factorisation and Broyden update are combined with Newton algorithm:

$$J_{l+k+1} = J_{l+k} + \frac{\Delta\Phi_{l+k} - J_{l+k} \Delta Z_{l+k}}{\|\Delta Z_{l+k}\|^2} \Delta Z_{l+k}^T \quad (2.16)$$

where $\Delta\Phi_{l+k} = \Phi_{l+k+1} - \Phi_{l+k}$ and $\Delta Z_{l+k} = Z_{l+k+1} - Z_{l+k}$. Moreover, when the first guess Z_0 is far from the the optimal solution, non-monotone iterations are used (86).

2.4 OCP for milling processes

In this study the OCP is formulated as a problem of Bolza, where the Lagrangian function is constituted by a linear combination of different terms taking into account the process productivity $J_p(x(\zeta), u(\zeta))$, quality $J_q(x(\zeta), u(\zeta))$, and cost $J_c(x(\zeta), u(\zeta))$. The dynamical system is controlled in force, therefore the control vector $u(\zeta)$ includes the linear acceleration of the axis $\dot{f}(\zeta)$ and the angular acceleration of the spindle $\dot{\omega}(\zeta)$. The vector of state variables $x(\zeta)$ includes the feed rate $f(\zeta)$ and the spindle speed $\omega(\zeta)$. Since several performances of the process need to be optimised, the OCP is formulated as a multi-objective minimisation:

$$\begin{aligned} J(u(\zeta), x(\zeta)) &= \int_{\zeta_0}^Z w_p J_p(x(\zeta), u(\zeta)) + w_q J_q(x(\zeta), u(\zeta)) + w_c J_c(x(\zeta), u(\zeta)) d\zeta + \\ &+ w_i \left(\frac{f(\zeta_i) - f_i}{f_{\min}} \right)^2 \end{aligned} \quad (2.17)$$

The Lagrange term is integrated from the current position of the mill ζ_0 to a future position Z . The Mayer term is imposed for safety reasons, thus ensuring that the optimal feed rate is not too different from the nominal one programmed by the user. Excessively high feed rates must be avoided since can damage the mill, or even break the spindle.

Each term of the target function is multiplied by a weight, thus contextualising the OCP to a specific application. In roughing operations the process productivity has the mayor contribute on the Lagrangian function, while process quality is more important during finishing cuts. The costs function is considered if hard metals are processed, since expensive tools might be used and high current consumptions might be required by long tool paths.

2.4.1 Productivity

The Material Removal Rate (MRR) quantifies the process productivity. High MRR are achieved by increasing the feed rate, as well as the depth, or width of the cut. The instantaneous MRR is calculated as:

$$MRR(\zeta) = cs(\zeta) f(\zeta) \quad (2.18)$$

where $cs(\zeta)$ is the instantaneous cut cross section. All the terms of the target function are normalised in order to have comparable values when all the weights in Eq. 2.17 are unitary. $J_p(x(\zeta), u(\zeta))$ is divided by the maximum MRR, which is given by setting the feed rate equal to the maximum feed of the machine axes Eq. 2.18. Moreover, since $J(u(\zeta), x(\zeta))$ has to be minimised, the inverse of the normalised MRR is used as descriptor of the process productivity:

$$J_p(x(\zeta), u(\zeta)) = \left(\frac{MRR(\zeta)}{MRR_{\max}(\zeta)} \right)^{-1} = \left(\frac{cs(\zeta) f(\zeta)}{cs(\zeta) f_{\max}(\zeta)} \right)^{-1} = \frac{f_{\max}(\zeta)}{f(\zeta)} \quad (2.19)$$

2.4.2 Quality

The quality of the process is defined by considering the roughness and the geometrical accuracy of the cut, which are affected by the cutting force acting and vibrations of the machine. The term associated to process quality is defined by three functions, which take into account: the deflection of the tool when a force is applied on it ($J_{q,p}(\zeta)$), as well as forced and self-excited vibrations ($J_{q,f}(\zeta)$ and $J_{q,c}(\zeta)$ respectively):

$$J_{q,p}(\zeta) = J_{q,p}(\zeta) + J_{q,f}(\zeta) + J_{q,c}(\zeta) \quad (2.20)$$

Deflection of the tool

Cutting forces acting on the tool are calculated by modelling the ram as a cantilever beam, in which the cutting force F_c is applied at the free end, Fig. 2.1. The deflection angle of the ram is:

$$\alpha = \frac{F_c(\zeta) l(\zeta)^2}{2 E J} \quad (2.21)$$

where E is the elastic modulus of the ram, J its inertia, and $F_c(\zeta)$ is the cutting force. Note that the length of the ram $l(\zeta)$ is dependent from the curvilinear abscissa since the W axis can be extended or retracted during the process. In order to compute F_c it

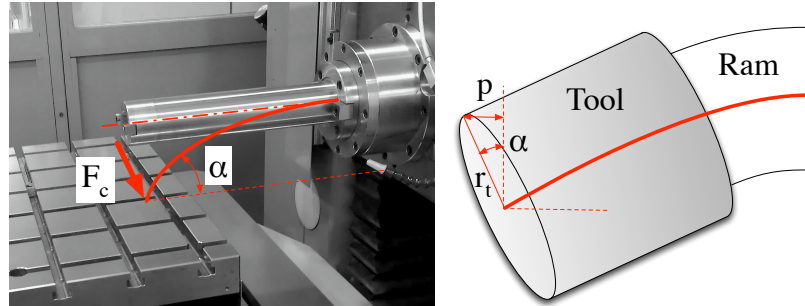


Figure 2.1: Model of the cantilever beam with the cutting force applied at the free end.

is necessary to estimate the cutting power P_c , which depends from the specific energy of the workpiece u_s :

$$P_c(\zeta) = cs(\zeta) f(\zeta) u_s = MRR(\zeta) u_s \quad (2.22)$$

$$F_c(\zeta) = \frac{P_c(\zeta)}{\omega(\zeta) r_t} \quad (2.23)$$

By knowing the angle α and the radius of the tool r_t it is calculated the penetration p of the tool into the workpiece that generates epicyclical streaks on the cut surface. The expression of p can be linearised since the tool is stiff and the deflection angle is usually small:

$$p(\zeta) = r_t \sin(\alpha) = r_t \alpha(\zeta) \quad (2.24)$$

$p(\zeta)$ is normalised on its maximum value, given by setting Eq. 2.21 equal to the maximum cross section cut along the tool path, while feed rate and spindle speed are respectively set to the maximum and minimum values supported by the drives. The expression of the function associated to the tool penetration that has to be minimised is:

$$J_{q,p}(\zeta) = \frac{p(\zeta)}{p_{\max}(\zeta)} = \frac{cs(\zeta) f(\zeta)}{\omega(\zeta)} \frac{\omega_{\min}}{cs_{\max} f_{\max}} \quad (2.25)$$

Self-excited vibrations

The vibrations of the machine deteriorate the quality of the process, by leaving periodic patterns on the cut surface. When analysing the dynamics of the cut, it is possible to distinguish between forced and self-excited vibrations.

The most dangerous self-excited vibrations are those regenerative, since as the cut proceeds the dynamics of the system becomes unstable, thus increasing progressively the chip thickness and consequently the magnitude of the cutting forces. The regenerative phenomena is due to a phase shift between vibration waves left on the inner and outer surfaces of the cut chip. Regenerative self-excited vibrations are called *chatter*, and its

manifestation is due to many factors that can be grouped into two main categories: process factors (material, tool geometry, lubrication, etc.) and dynamic factors (tool-workpiece relative dynamic compliance). The relationship between stability occurrence and the above-mentioned factors can be analysed through a reduced set of variables by means of two main approaches: the multi-frequency and the single-frequency 0-order approach (*ZOA*). For non-interrupted milling processes, the single-frequency model allows a fast and accurate calculation of the stability lobes. Due to the fact that heavy-duty milling operations are usually performed with large (75-80%) tool engagements and high number of flutes, the created harmonics are usually weak, and the single-frequency method can provide an accurate solution (87). Adopting this latter approach, the stability analysis is traced back to the following characteristic equation of the dynamic system “machine tool+milling process” (see Appendix B):

$$\det \left(\mathbf{I} + \Lambda \mathbf{A}_0(\zeta) \hat{\mathbf{G}}_{tool-wp}(s|\zeta) \right) = 0 \quad (2.26)$$

with

$$\Lambda = -\frac{n_t}{4\pi} d K_t \left(1 - e^{-(\sigma+i\omega)\frac{2\pi}{\Omega n_t}} \right) \quad (2.27)$$

where d is the depth of cut, n_t is the number of tool cutters, $(\sigma + i\omega)$ is the pole of the dynamical system, and Ω is the spindle speed in rad/s. \mathbf{A}_0 is a matrix that takes into account the orientation of the average cutting force w.r.t. feed and normal axes, and depends on tool engagement condition together with tangential and radial cutting pressures (hereafter K_t and K_r , respectively). $\hat{\mathbf{G}}_{tool-wp}$ is the relative dynamic compliance between tool and workpiece in the complex variable $s = \sigma + j\omega$, defined w.r.t. the same reference frame of \mathbf{A}_0 . Without lose of generality, since in our application the workpiece is rigid, only the tool compliance is taken into consideration, namely, $\hat{\mathbf{G}}_{tool-wp} \simeq \hat{\mathbf{G}}_{tool}$.

Imposing $\sigma = 0$ and scanning the chatter frequency ω , the Eq. (2.27) can be used to

map the stability limit w.r.t. depth of cut and spindle speed. The result of this operation is the well-known *stability lobes diagram* (SLD). On the other side, given a depth of cut and a spindle speed, Eqs. (2.26) and (2.27) can be solved w.r.t. σ and ω . Note that σ expresses the exponential growth of vibration during unstable cut and can be used as a metric to measure the instability/stability level. Unfortunately, this equation system is transcendent and the solution can be found only through time-consuming numerical algorithms. A viable heuristic solution consists in computing the SLD and in evaluating the instability level by the difference between the actual depth of cut and the limiting one. In this sense, the SLD itself is used to generate a the penalty function relating instability level to spindle speed.

In Eq. (2.26) the compliance matrix $\hat{\mathbf{G}}_{tool}$ depends on curvilinear abscissa not only in terms of tool position along the tool path, but also in terms of feed velocity direction. The latter can be expressed by applying a rotation to the dynamic compliance measured w.r.t. the global machine axes:

$$\hat{\mathbf{G}}_{tool}(\zeta|\omega) = \mathbf{R}(\mathbf{v}) \mathbf{G}_{tool}^{axes}(\zeta|\omega) \quad (2.28)$$

with

$$\mathbf{R}(\mathbf{v}) = \begin{bmatrix} v_x & -v_y \\ v_y & v_x \end{bmatrix} \quad (2.29)$$

The dynamic compliance at tool tip must be measured, or estimated, for a given tool and for a suitable number of position samples along the tool path, wherever SLDs have to be computed. Since the measurement of the necessary set of dynamic compliances may be quite onerous, a viable makeshift consists in regressing an adequate mesh of primitive dynamic compliances covering the available workspace (as explained in 6.3). On the other side, the response at tool tip can be derived from that at spindle nose by exploiting the well-known receptance coupling sub-structuring technique (88): in this way the measurements do not have to be repeated whenever the tool is changed.

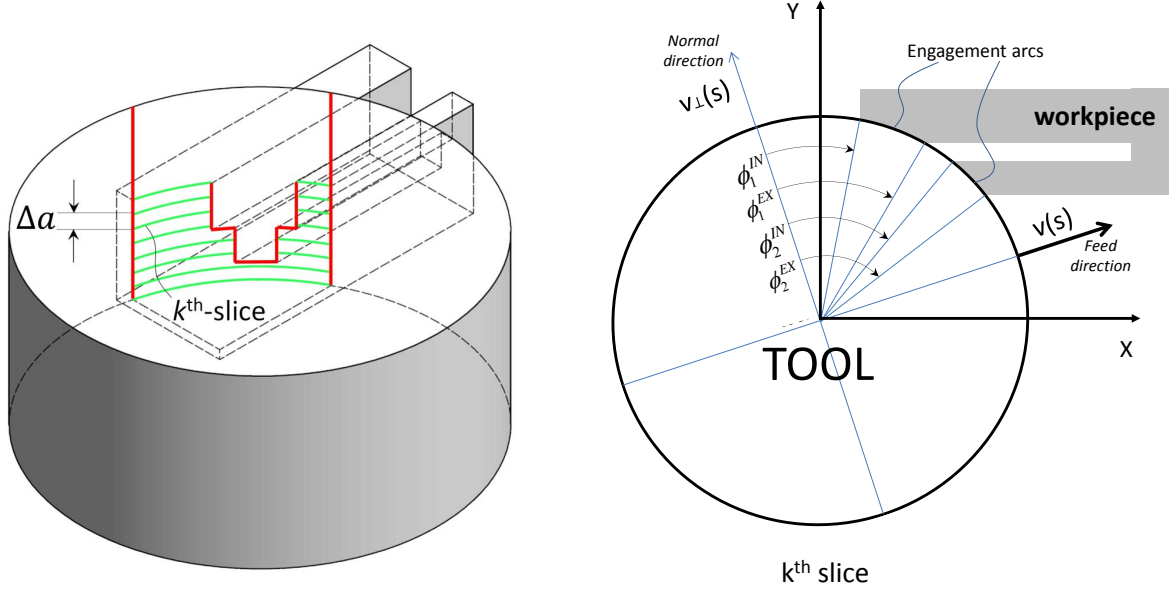


Figure 2.2: Engagement arcs definition w.r.t. feed velocity.

The orientation matrix \mathbf{A}_0 relates the vibrational displacement of the tool along feed and normal direction to the average cutting force vector due to the consequent material removal. The matrix can be computed by integrating the forces exerted by each cutter engaged in the workpiece. If the cylinder bounding the cutters is considered, tool engagement can be described by the arcs of contact between this cylinder and the workpiece. Generally speaking, the arcs of contact may be more than one for each tool section, and may vary along the tool axis (see Fig.2.2). In order to deal with such inherent complexity, the list of the contact arcs for each tool slice is computed by a dedicated code that exploits commercial 3D CSG libraries (*ACIS*, developed by *Spatial* [®] Corporation). Once the tool engagement is properly described, the matrix

\mathbf{A}_0 is computed as it follows (i.e., the generalisation of the corresponding quantity presented in (53)):

$$\mathbf{A}_0 = \begin{bmatrix} a_{ff} & a_{fn} \\ a_{nf} & a_{nn} \end{bmatrix} \quad \text{with} \quad \begin{aligned} a_{ff} &= \frac{1}{a_{tot}} \sum_k \Delta a \sum_i \frac{1}{2} (\cos 2\phi - 2K_r \phi + K_r \sin 2\phi) \Big|_{\phi_{k_i}^{IN}}^{\phi_{k_i}^{EX}} \\ a_{fn} &= \frac{1}{a_{tot}} \sum_k \Delta a \sum_i \frac{1}{2} (-\sin 2\phi - 2\phi + K_r \cos 2\phi) \Big|_{\phi_{k_i}^{IN}}^{\phi_{k_i}^{EX}} \\ a_{nf} &= \frac{1}{a_{tot}} \sum_k \Delta a \sum_i \frac{1}{2} (-\sin 2\phi + 2\phi + K_r \cos 2\phi) \Big|_{\phi_{k_i}^{IN}}^{\phi_{k_i}^{EX}} \\ a_{nn} &= \frac{1}{a_{tot}} \sum_k \Delta a \sum_i \frac{1}{2} (-\cos 2\phi - 2K_r \phi - K_r \sin 2\phi) \Big|_{\phi_{k_i}^{IN}}^{\phi_{k_i}^{EX}} \end{aligned} \quad (2.30)$$

where a_{tot} is the total tool depth engaged in the workpiece, Δa is the tool discretisation spacing in axial direction, $\phi_{k_i}^{IN}$ and $\phi_{k_i}^{EX}$ are respectively the initial and the exit angle of the i^{th} arc of the k^{th} tool slice.

The approach presented in this study is illustrated in Fig. 2.3 where SLDs are calculated at each sampled curvilinear abscissa. At each position of the tool, the stability limit d_{lim} , and position of the highest peak d_{max} in SLD are extracted. Then, the

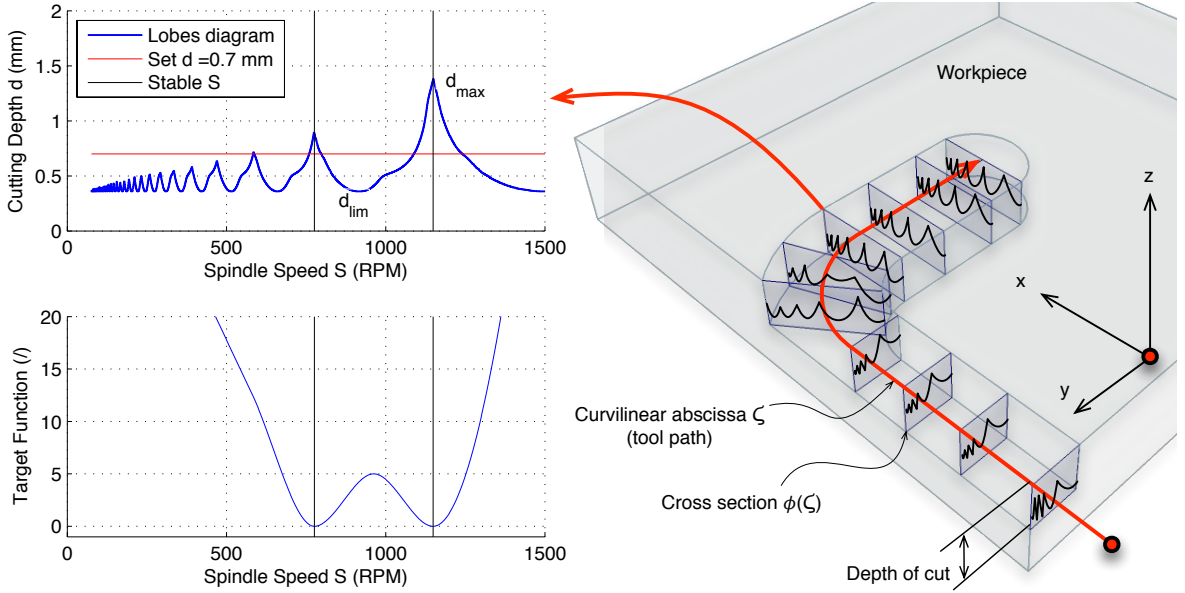


Figure 2.3: Discretisation of the tool path (right) and computation of the target function at each quantised position increment (left).

term $J_{q,c}(\zeta)$ of the target function necessary to mitigate the chatter is obtained from the intersection of the SLD with the actual cutting depth $d(\zeta)$. Three conditions are identified:

1. $d(\zeta) < d_{lim}$: $J_{q,c}(\zeta)$ is zero since all the spindle speed allow a stable cut.
2. $d(\zeta) > d_{max}$: it is not possible to achieve a stable cut, but in order to limit unavoidable chatter, $J_{q,c}(\zeta)$ is formulated as a cubic spline with the minimum at the spindle speed corresponding to d_{max} .
3. $d_{lim} \leq d(\zeta) \leq d_{max}$: the abscissa coordinate of the stability pockets are identified and used to generate a set of nodes that are interpolated with a Piecewise Cubic Hermite Polynomial (89). Peaks close to actual cutting depth less than 0.1 mm are neglected in order to deal with the uncertainty of the model parameters used to compute the stability lobes diagrams. Auxiliary nodes with non-null ordinate are placed between the stable spindle speeds to avoid instability pockets. In the end, the minimum and maximum point are characterised by defining each node as a stationary point.

A similar approach was followed by Merdol and Altintas (90). The authors developed a process optimisation system to calculate the cutting speed, feed rate, depth and width of the cut which maximised the MRR, while considering the dynamics of the process along the tool path. This methodology however did not take into account the effect of forced vibrations, and it was tested only on peripheral milling operations.

It is worth to be remarked that ZOA approach assumes that \mathbf{A}_0 and $\hat{\mathbf{G}}_{tool}(\omega|\zeta)$ do not depend on time, while, as matter of fact, this assumption does not hold in the case of complex tool trajectories generated by a general part program, since tool engagement varies continuously as well as structural machine dynamics. However, the assumption

can be considered still valid on the basis of the so-called frozen time approach based on the "Adiabatic Theorem", also adopted to tackle variable time delay systems (91). It states: *A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the system spectrum.* (92). If, in a time window ΔT , the variation of the product $\mathbf{A}_0 \hat{\mathbf{G}}_{tool}$ is negligible and ΔT is much greater than the tooth passing period, the system dynamics (whose pole is given by $(\sigma + i\omega) / \tau$) is much more faster than the perturbation action. The time-varying $\mathbf{A}_0 \hat{\mathbf{G}}_{tool}$ can then be assumed to be frozen in the interval ΔT , i.e.:

$$\mathbf{A}_0 \hat{\mathbf{G}}_{tool} (\zeta(t)) \approx \text{const.} \quad \text{for } t \in \Delta T \quad (2.31)$$

Since the chatter frequency is a function of spindle speed, the variability of $\hat{\mathbf{G}}_{tool}$ must be evaluated for a suited range of frequency, namely, in the neighbourhood of the resonance that are candidate to be source of chatter problems. Algorithmically, the following expression is proposed to verify the consistency of the frozen time assumption:

$$\frac{\|\mathbf{A}_0(\zeta(t_{k+1})) \hat{\mathbf{G}}_{tool}(\omega, \zeta(t_{k+1})) - \mathbf{A}_0(\zeta(t_k)) \hat{\mathbf{G}}_{tool}(\omega, \zeta(t_k))\|}{\|\mathbf{A}_0(\zeta(t_k)) \hat{\mathbf{G}}_{tool}(\omega, \zeta(t_k))\|} < \epsilon, \quad (2.32)$$

$$\forall \omega \in F_c \quad \text{and} \quad t_{k+1} - t_k > M \tau_{TPF}$$

where $\zeta(t_k)$ and $\zeta(t_{k+1})$ are two subsequent curvilinear abscissas, τ_{tpf} is the tool passing frequency, ϵ and M are two parameters to be set after proper experiments, and F_c is the union of neighbourhoods of the relevant tool tip resonances. In case the (2.32) is not verified, the stability analysis is unreliable and the corresponding term in the global objective function must be neglected.

The last issue to be tackled concerns the rate of the stability analysis that must be performed along the curvilinear abscissa during the process simulation. In order to increase the computational efficiency, the SLDs should be updated and stored only if necessary. Only the eigenvalue Λ varies along the tool path (due to changes in

tool engagements and dynamics), this quantity itself can be used to drive the SLDs evaluation density. The SLD is then re-calculated and stored if the eigenvalue change in two consecutive positions is more than a given threshold:

$$\frac{1}{N} \sum_i |\Lambda_k(\omega_i) - \Lambda_{k+j}(\omega_i)| > \epsilon \quad \text{with } \omega_i \in \{\omega_1, \omega_2, \dots, \omega_N\} \quad (2.33)$$

where k refers to the last abscissa sample corresponding to a new SLD, j is the sample lapse till the current abscissa, ω_i are the chatter frequency samples on which Λ is calculated, and ϵ is a parameter to be set on the basis of a trade-off between precision and computational power. Finally, SLDs are used only in a spindle speed range which is feasible from a technological point of view.

Forced vibrations

In spite of process stability, even forced vibrations could be significant and detrimental, so they must be considered in the optimisation objective as well. Forced vibration level is evaluated by considering the response of the tool tip to the dynamic components of cutting force, that can be computed from tool engagement and cutting parameters (in a similar way than the average component, as explained in (93)).

Starting from the ideal cutting mechanics, the dynamic cutting force components in frequency domain are harmonics of the tool passing frequency. The r^{th} harmonic can be written as it follows:

$$\mathbf{f}_{dyn}^{(r)}(\zeta) = \frac{1}{4\pi} K_t \frac{a_{tot}}{N(\zeta)} f(\zeta) \begin{Bmatrix} a_{ff}^{(r)} \\ a_{nf}^{(r)} \end{Bmatrix} \quad (2.34)$$

with

$$\begin{aligned} a_{ff}^{(r)} &= \frac{1}{a_{tot}} \sum_k \Delta a \sum_i \frac{j}{2} \left(-c_0 K_r e^{-jrN_{cut}\phi} + c_1 e^{-jp_1\phi} - c_2 e^{jp_2\phi} \right) \Big|_{\phi_{k_i}^{IN}}^{\phi_{k_i}^{EX}} \\ a_{nf}^{(r)} &= \frac{1}{a_{tot}} \sum_k \Delta a \sum_i \frac{1}{2} \left(c_0 e^{-jrN_{cut}\phi} + c_1 e^{-jp_1\phi} + c_2 e^{jp_2\phi} \right) \Big|_{\phi_{k_i}^{IN}}^{\phi_{k_i}^{EX}} \end{aligned} \quad (2.35)$$

where $p_1 = 2 + rN_{cut}$, $p_2 = 2 - rN_{cut}$, $c_0 = \frac{2}{rN_{cut}}$, $c_1 = \frac{K_r - j}{p_1}$ and $c_2 = \frac{K_r + j}{p_2}$, with j denoting the imaginary unit (the other symbols have been defined after (2.30)). The number of relevant harmonics (r) of the tooth passing frequency (ω_{tpf}) to be considered for an accurate reconstruction of the dynamic cutting force components depends on the immersion conditions and on the number of teeth in the cut.

Finally, the vibration index can be identified with the magnitude of the resulting tool tip vibrational displacement:

$$F_{forced} = \left\| \sum_{r \in ([-R, R] - \{0\})} \hat{\mathbf{G}}_{tool}(r\omega_{TPF}|\zeta) \cdot \mathbf{f}_{dyn}^{(r)} \right\| \quad (2.36)$$

where R is the number of harmonics to be considered (usually $3 \div 5$).

2.4.3 Costs

In a constant cutting-speed, rough-machining operation, the production cost C_{pr} per component can be calculated by (94):

$$C_{pr} = C_m + M t_l + M t_m + (M t_{ct} + C_t) \frac{t_m}{t} \quad (2.37)$$

where C_m is the material cost, M is the total machine and operator rate, t_l is the nonproductive time, t_m is the machining time, t_{ct} is the tool changing time, and C_t is the cost of providing a sharp tool. The minimum cost of production can be calculated

for each operation, through Taylor's tool life equation:

$$C = v_c t^n \quad (2.38)$$

$$v_c = 2 \pi r_t s = 0.06 \omega r \quad (2.39)$$

where C is a constant, v_c is the cutting speed, s and ω is the spindle speed in rpm and rad/s respectively, r_t is the radius of the tool, n depends on the cutting conditions, and t is the cutting time. The machining time t_m is:

$$t_m = \frac{K}{v_c} \quad (2.40)$$

K is a constant, different for each operation, that can be interpreted as the distance traveled by the tool cutting edge corner relative to the workpiece. By substituting Eqs. 2.38 and 2.40 into Eq. 2.37, after some calculations that are omitted for brevity, the minimum cost of production C_{\min} is:

$$C_{\min} = M t_l + \frac{M K}{(1 - n) v_r} \left(\frac{t_c}{t_r} \right)^n \quad (2.41)$$

C_{\min} depends on the type of the operation carried out, as well as on the tool wear. The former is a constant that cannot be optimised if not by redesigning the whole process. Tool wear instead can be reduced with an accurate selection of process parameters. Taylor's wear model has been formulated mainly for turning operations, and it is too simple for milling processes since it does not take into account the cut cross section. The wear model 2.38 is therefore extended, and tool wear wt is calculated from the multiplicative inverse of the tool life t_{tl} , that is expressed in minutes:

$$t_{tl}(\zeta) = n_t \left(\frac{C}{v_c(\zeta) f_z(\zeta)^a c_s(\zeta)^b} \right)^{\frac{1}{n}} \quad (2.42)$$

$$f_z(\zeta) = \frac{f(\zeta)}{n_t s(\zeta)} = \frac{2\pi}{60} \frac{f(\zeta)}{n_t \omega(\zeta)} \quad (2.43)$$

$$wt(\zeta) = \frac{1}{t_{tl}(\zeta)} \quad (2.44)$$

where n_t is the number of flutes in the mill, $f_z(\zeta)$ is the instantaneous feed per tooth, while C , n , a and b are the tool's wear coefficients. $wt(\zeta)$ is normalised on the maximum tool wear, given when the spindle speed is minimum, and the cross section together with feed rate are maximum. The final expression of the normalised tool wear that has to be minimised is:

$$J_{c,w}(\zeta) = \frac{wt(\zeta)}{wt_{\max}} = \left[\frac{\omega(\zeta) r_t}{\omega_{\max} r_t} \left(\frac{f(\zeta)}{n_t \omega(\zeta)} \frac{n_t \omega_{\max}}{f_{\max}} \right) \left(\frac{cs(\zeta)}{cs_{\max}} \right)^b \right]^{\frac{1}{n}} \quad (2.45)$$

The model in Eq. 2.37 lacks in considering the energy spent during the process. Energy consumption is a factor with respect to which CNC users have grown interests in the last years, due to the raise of energy price and more strict environmental regulations. During a cut, electric current is mainly absorbed by the spindle, which rotates at high speeds. Electric energy absorbed by a motor is proportional to the kinetic energy developed. It means that the rotational kinetic energy of the spindle can be used as an estimator of energy consumption:

$$E_e \propto E_k = \frac{1}{2} I \omega^2 \quad (2.46)$$

J is the moment of inertia of all the kinematic chain from the tool to the electric drive. Once E_k it is normalised on the maximum kinetic energy developable by the spindle, the target function associated to energy consumption is defined as:

$$J_{c,e}(\zeta) = \frac{E_e}{E_{\max}} = \left(\frac{\omega(\zeta)}{\omega_{\max}} \right) \quad (2.47)$$

Finally, Eqs. 2.45 and 2.47 are summed to form $J_c(\zeta)$:

$$J_c(\zeta) = J_{c,w}(\zeta) + J_{c,e}(\zeta) \quad (2.48)$$

2.4.4 DAE constraints

Algebraic constraints are imposed in order to calculate controls whose values are within the performance of the CNC machine, and respect safety conditions. One sided constraints take into account the maximum mechanical torque and power generated by the spindle:

$$\begin{bmatrix} T_c(x(\zeta), u(\zeta)) \\ P_c(x(\zeta), u(\zeta)) \end{bmatrix} \leq \begin{bmatrix} T_{\max} \\ P_{\max} \end{bmatrix} \quad (2.49)$$

where $T_{\max} = 57.3 \text{ Nm}$, and $P_{\max} = 11 \text{ kW}$. A third inequality is added to Eq. 2.49 in order to limit the roughness of the cut. The maximum feed rate is calculated through an heuristic relationship taking into account the tool fillet radius $r_f = 0.8 \text{ mm}$. This equation is suggested by the tool manufacturer:

$$f(\zeta) \leq \frac{0.6}{2\pi} r_f \omega(\zeta) \Phi(\zeta) \quad (2.50)$$

the $\Phi(\zeta)$ function is zero for null cut cross sections, otherwise its value is one. By doing so, it is possible to include this extra constraint without affecting the process productivity during in-air fast movements.

Two sided constraints are set to avoid excessively high feed rates and spindle speeds, but also to void dangerous situations, such as a non rotating mill impacting against

the workpiece:

$$\begin{bmatrix} \omega_{\min} \\ f_{\min} \end{bmatrix} \leq \begin{bmatrix} \omega(\zeta) \\ f(\zeta) \end{bmatrix} \leq \begin{bmatrix} \omega_{\max} \\ f_{\max} \end{bmatrix} \quad (2.51)$$

where $\omega_{\min} = 10.5 \text{ rad/s}$, $\omega_{\max} = 157 \text{ rad/s}$, $f_{\min} = 0 \text{ mm/min}$, and $f_{\max} = 10000 \text{ mm/min}$.

The last set of constraints included into the OCP take into account the dynamics of the machine. From preliminary tests, it has been observed that the dynamics of the state variables is fast, instead the controls are limited in acceleration. Feed rate and spindle speed velocity are modelled as a first order system:

$$\begin{bmatrix} f(\zeta) \frac{d}{d\zeta} f(\zeta) - v f(\zeta) \\ f(\zeta) \frac{d}{d\zeta} \omega(\zeta) - v \omega(\zeta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.52)$$

Chapter 3

Simulacrum

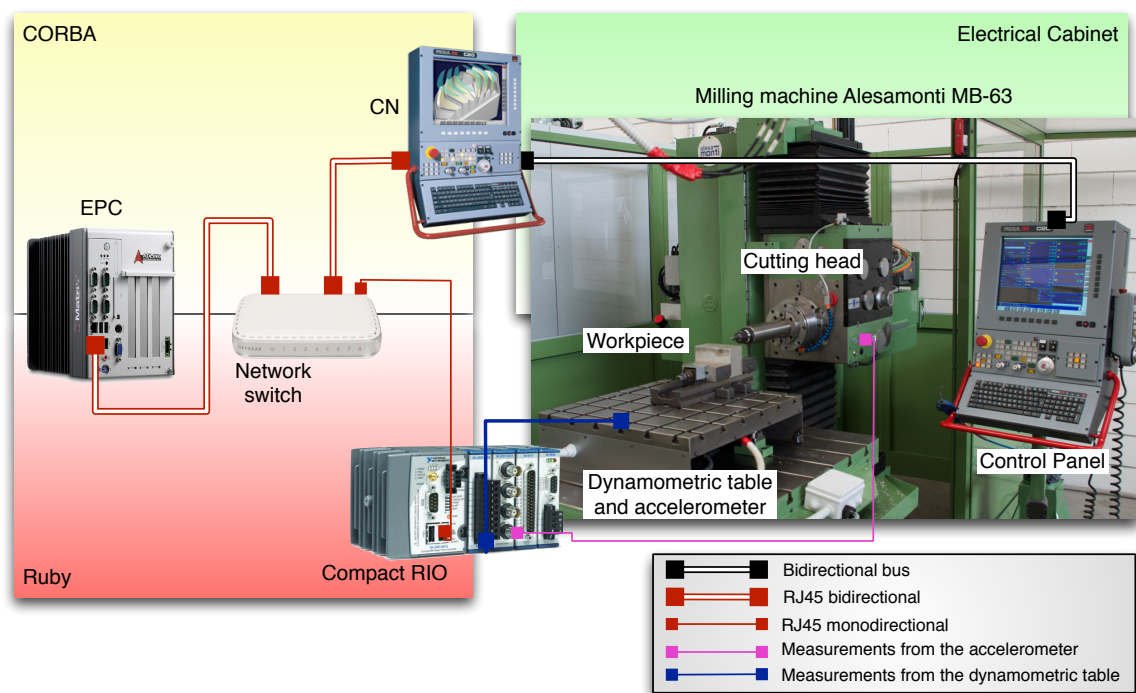


Figure 3.1: Architecture of the simulacrum.

The simulacrum developed to test and evaluate the Evaluation and Perception Controller (EPC) is schematised in Fig. 3.1. The system is constituted by three elements:

NC milling machine, measurement system, and EPC that is installed on an industrial fan-less computer with a Dual Core Atom 1.6 GHz CPU. A Local Area Network (LAN) has been created in order to allow the communication between the three subsystems of the simulacrum. The state of the process is reconstructed through the data recorded by the NC and by the CompactRIO. The latter is used to acquire a triaxial accelerometer on the cutting head, and a triaxial dynamometric table on which the workpiece is clamped. EPC and NC are communicating through a bi-directional connection, which is bridged by the user panel of the machine.

Each hardware and software component of the system is described in this chapter, with exception of the communication layer which will be outlined and characterised in Chapter 5.

3.1 Milling machine Alesamonti MB63

The five-axis milling machine *Alesamonti MB63* is used as simulacrum to test and validate the EPC. According to the norm *ISO 3070 part I*, the MB63 is classified as a machine with a fixed column and movable table, Fig. 3.2. The technical drawings of the structure are reported in Appendix E. Movements along X and Z axes are associated to the working table and workpiece. The cutting head is moved vertically along the Y axis, while B and W are used to turn the working table and to extract the ram respectively. The three linear axes are moved through recirculating ball screws, and they are actuated by brushless motors. Digital drives *Fidia XPower* are used to control the electric motors of axes and spindle. The technical specifications of the machine are summarised in Tabs. 3.1 and 3.2. The kinematic chain of the cutting head in Fig. 3.3 is analysed to characterise the cutting torque at the spindle (as it will be explained in Chapter 6.2.2). In Tab. 3.3 are reported the geometrical information of all the shafts.

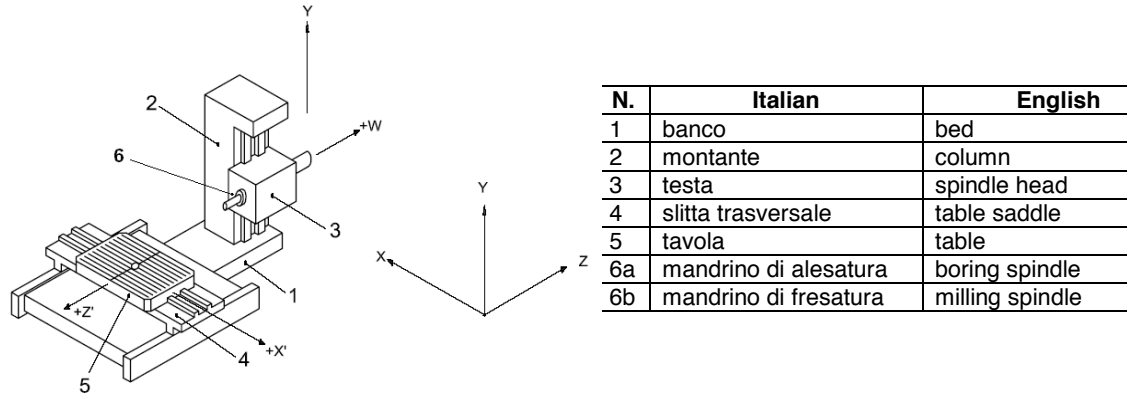


Figure 3.2: Configuration of the machine MB63.

Table 3.1: Technical specifications of the MB63.

Data	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>W</i>	<i>B</i>
<i>Travel</i>	729 mm	556 mm	734 mm	369 mm	360 °
<i>Max. speed</i>	10000 mm/min	10000 mm/min	10000 mm/min	5000 mm/min	720 °/min
<i>Stall torque of the motor</i>	20 Nm	13 Nm	20 Nm	13 Nm	20 Nm
<i>Max. speed of the motor</i>	3000 rpm	3000 rpm	3000 rpm	3000 rpm	3000 rpm

Table 3.2: Technical specifications of the spindle.

Data	Value
<i>Brand</i>	Siemens
<i>Model</i>	1PH7107-2NF02-0CA3
<i>Max. speed</i>	1500 rpm
<i>Power in S6 (60%)</i>	11 kW
<i>Constant torque of the torque</i>	57.3 Nm
<i>Constant torque in high range</i>	72,77 Nm
<i>Cone type</i>	ISO 40 DIN 2080
<i>Diameter of the boring spindle</i>	70 mm

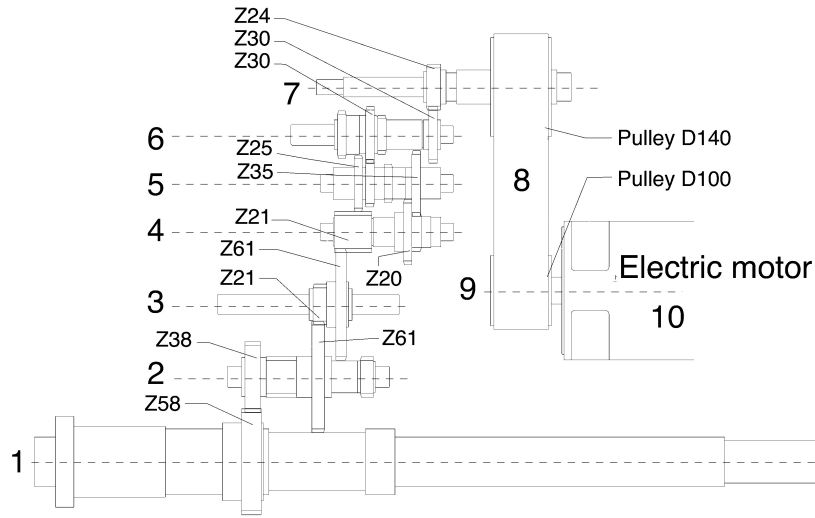


Figure 3.3: Kinematic chain of the cutting head.

Table 3.3: Technical specifications of the shafts in the kinematic chain, Z is the number of teeth of the gear connected to the shaft, and D is the diameter of the pulley. The missing information can be extracted from the CAD drawings in Appendix E.

Number	Description	Material
1	spindle with gear, $Z = 58$	Steel 18 NiCrMo5
2	Shaft, $Z_1 = 38$ and $Z_2 = 61$	Steel 18 NiCrMo5
3	Shaft, $Z_1 = 21$ and $Z_2 = 61$	Steel 18 NiCrMo5
4	Shaft, $Z_1 = 21$ and $Z_2 = 20$	Steel 18 NiCrMo5
5	Shaft, $Z_1 = 35$ and $Z_2 = 25$	Steel 18 NiCrMo5
6	Shaft, $Z_1 = 30$ and $Z_2 = 30$	Steel 18 NiCrMo5
7	Shaft, $Z_1 = 24$ and $D_2 = 140$ mm	Steel 18 NiCrMo5 and GG25 DIN 1691
8	V-belt, long 375 mm and width 16 mm	polyamide
9	Pulley $D = 100$ mm	GG25 DIN 1691
10	Electric motor 1PH7107-2NF01-0CA3	See data sheet

3.2 Numerical Control FIDIA C20

The numerical control installed on the simulacrum is the *FIDIA C20*. The architecture of this unit is characterised by two processors dedicated to different tasks: a RISC Power PC is utilised for axis and PLC management, while the PC side is aimed at managing the user interface. Thanks to this architecture it is possible to perform even onerous operations on the user panel without interfering with the ongoing machining.

3.3 Sensors

The process is sensed not only through the information available in the NC, but also through a dynamometric table and a triaxial accelerometer, Fig. 3.4. These sensors are acquired through a *National Instrument CompactRIO cRIO 9012* with a reconfigurable chassis *NI 9103*. The cRIO has a FPGA module which is used to sample at high frequency (5 kHz) the input/output modules to which the sensors are connected. The accelerometer is plugged to a *NI 9234* module, while the dynamometric table is connected to a *NI 9215* module. The specifications of the sensors are summarised in the next sections.

The collected information is elaborated, and sent to the EPC at a low frequency (5 Hz), in order to have a communication that is synchronous with the optimisation loop running at 5 Hz. The software loaded on the cRIO has been developed by the group of mechatronic engineering of the university of Trento.

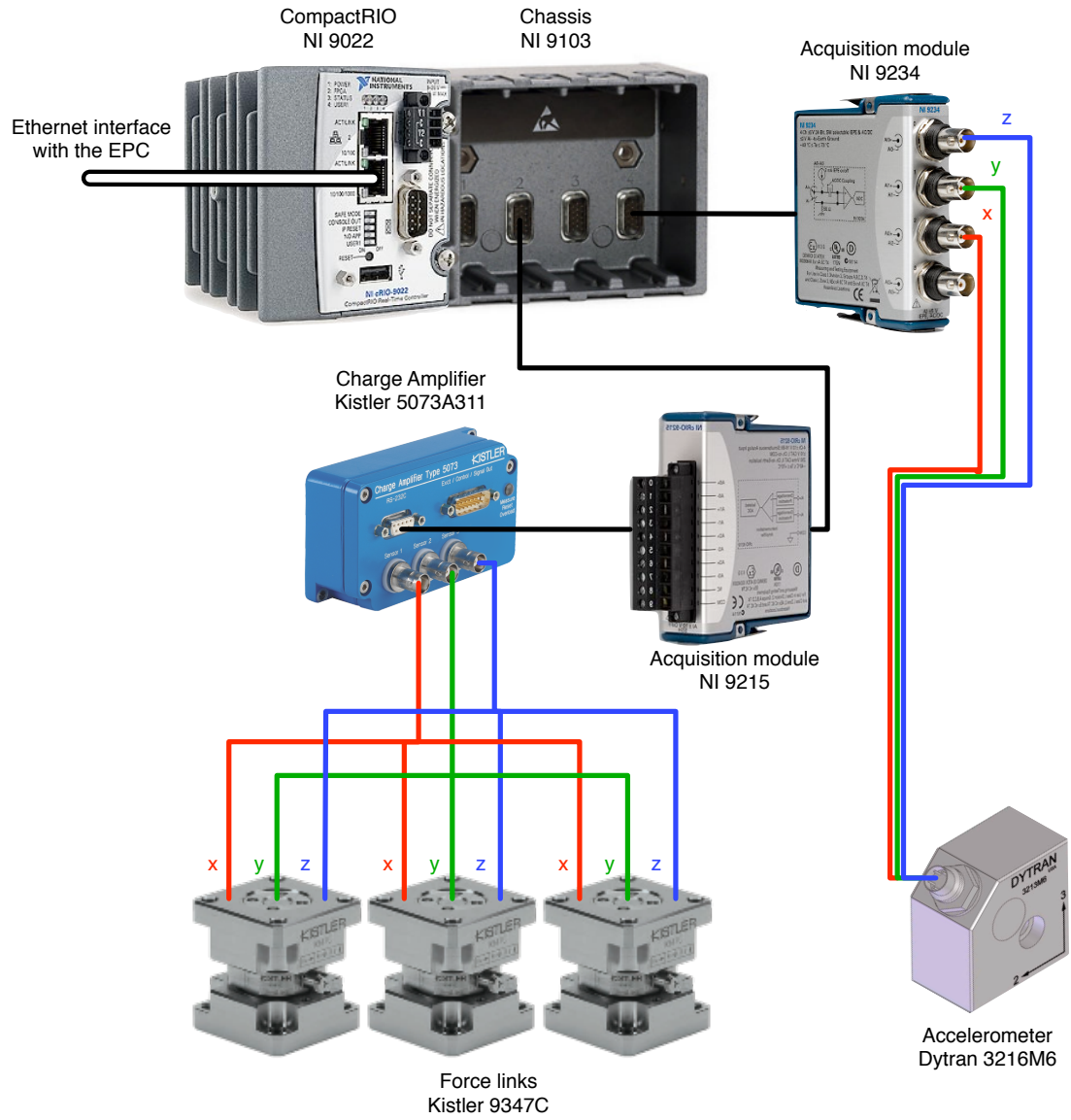


Figure 3.4: Scheme of the measurement system.

3.3.1 Dynamometric table

The dynamometric table consists of three piezoelectric *Kistler 9347C* 3-component force links on top of which is fixed a customised working table, Fig. 3.5. The specifications of the working table, and of the force links are summarised in Tab. 3.4 and in Tab. 3.5 respectively. A charge amplifier *Kistler 5073A311* is used to convert the charge measured by the piezoelectric sensors into a voltage signal, Tab 3.6. Note that the three components of the force measured by each dynamometer are analogically summed, thus reducing the number of acquired channels to three. By doing so it is possible to cut the cost of the system, since only one charge amplifier and one NI 9215 module are needed.

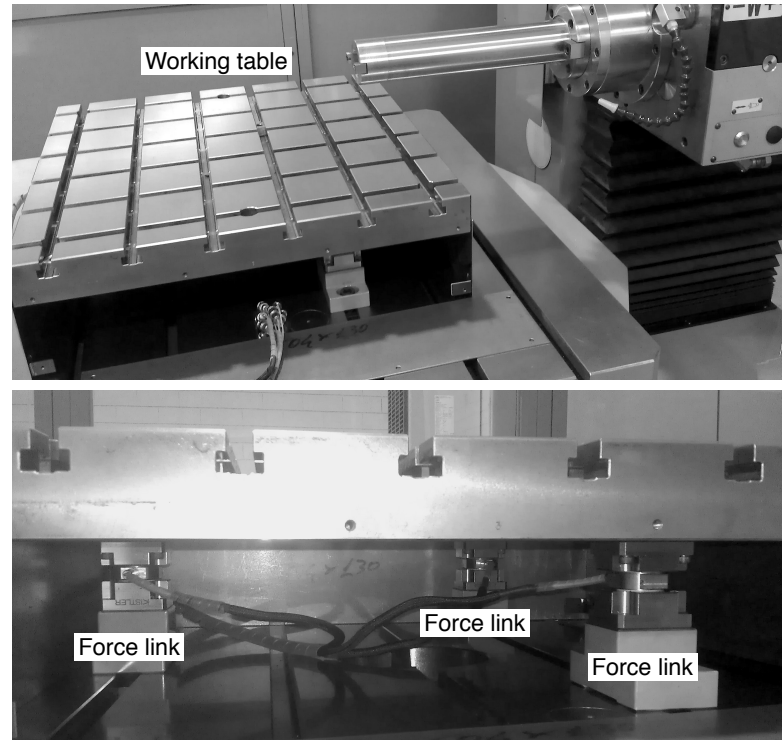


Figure 3.5: The dynamometric table.

Table 3.4: Technical specifications of the working table.

Parameter	Value
<i>Weight</i>	300 kg
<i>Dimensions [L x W x H]</i>	600 mm x 600 mm x 60 mm
<i>Width of T-slots</i>	14 mm
<i>Flatness tolerance</i>	0.01 mm
<i>Parallelism tolerance between upper and lower planes</i>	0.01 mm
	0.02 mm

Table 3.5: Technical specifications of the piezoelectric force links Kistler 9347C.

Parameter	Value
<i>Range along X and Y</i>	± 15 kN
<i>Range along Z</i>	± 30 kN
<i>Overload</i>	10%
<i>Sensibility along X and Y</i>	-7.8 pC/N
<i>Sensibility along Z</i>	-3.7 pC/N
<i>Linearity</i>	$\leq \pm 0.5$ % FSO

Table 3.6: Technical specifications of the charge amplifier Kistler 5073A311.

Parameter	Value
<i>Range</i>	± 100 pC, $\pm 1,000,000$ pC
<i>Output voltage</i>	0 V, ± 10 V
<i>Maximum output current</i>	± 5 mA
<i>Measurement accuracy</i>	$\leq \pm 0.5$ %

3.3.2 Accelerometer

The technical specifications of the Integrated Electronic PiezoElectric (IEPE) three-axial accelerometer *Dytran 3213M6* are reported in Tab. 3.7.

Table 3.7: Technical specifications of the accelerometer Dytran 3213M6.

Parameter	Value
<i>Range</i>	$\pm 50\text{ g}$
<i>Sensitivity $\pm 10\%$</i>	100 mV/g
<i>Frequency response $\pm 10\%$</i>	1.5 to 5000 Hz
<i>Linearity</i>	$\pm 1\%$ FSO
<i>Maximum trasversal sensitivity</i>	5%

Chapter 4

Evaluation and Perception Controller

The Evaluation and Perception Controller is constituted by three elements: a simulator, the optimisation library, and the perception layer (Fig. 4.1). In the process simulator a simplified Numerical Control (RNC) analyses the tool path. The G-Code can be generated either manually or through a CAM software, then it is loaded and parsed by the RNC. This software calculates also the feed profiles for every block in the G-Code (i.e. G01, G02, and G03), and interpolates the tool trajectory with the desired sample time. The parsed tool path, together with a model of the kinematic of the machine, workpiece and tool geometries, is fed into a geometric kernel in order to compute the cut cross section, cutting forces and stability lobes diagram along the curvilinear abscissa. The simulation of the cut is the most time consuming task since it requires the continuous update of the workpiece geometry at each simulated tool position. Once all the blocks of G-code are simulated, the generated models are used to perform an offline optimisation, in which the optimal feed rate and spindle speed profiles are computed by solving an Optimal Control Problem. When the optimisation

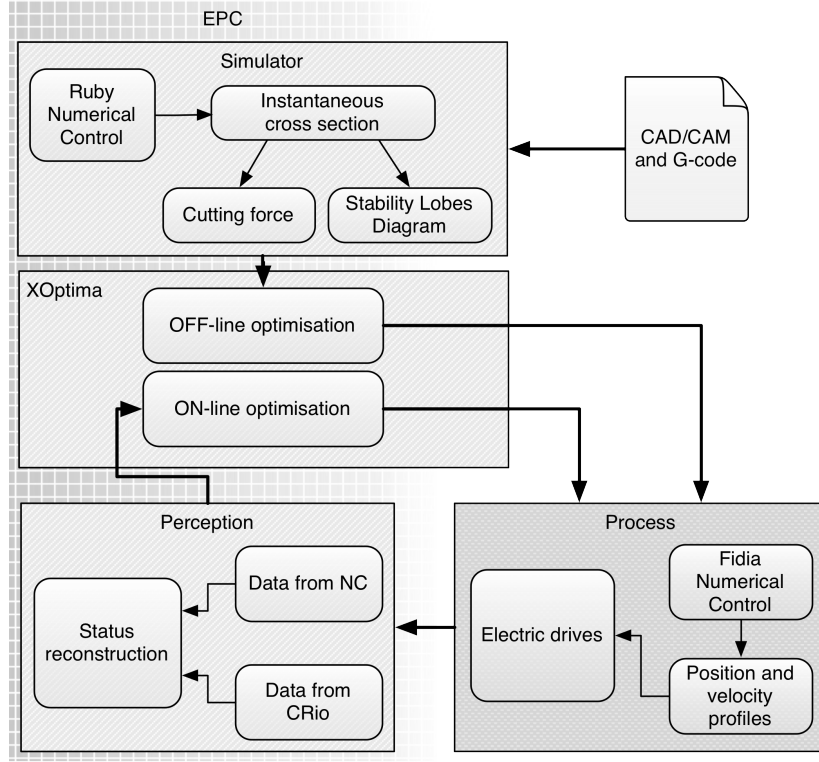


Figure 4.1: Architecture of the EPC.

converges, the process is started and the nominal controls programmed into the G-code are overridden with the pre-optimised ones. During the cut, the data recorded by the NC and sensors in the working area are utilised by a perception layer to reconstruct and learn the state of the process. By having the current status of the system, it is possible to maintain constantly updated the models of the process and to re-optimize online the controls. Through the online loop, deviations of the process state from the predictions made during the offline optimisation can be compensated. As instance, it is possible to correct the controls profiles by estimating online the specific energy of the cut material, whenever a workpiece is harder (or even softer) than expected. It must be remarked that the offline optimisation is essential to have an online adaptive control, since the pre-calculated optimal control ease the convergence of the online optimisation. The offline optimal controls are used as a first guess during the online

optimisation, and are applied to the system whenever the online optimisation struggles to converge.

The EPC is implemented as a state machine, whose architecture is schematised in Fig. 4.2. This code structuration allows the software to be highly flexible, expandable, and easily maintainable. Whenever a new functionality has to be added to the system, it is only necessary to implement a new state and to re-define the tree structure of the state machine by specifying the dependences between the new state and the already existing ones. The implemented states are the following:

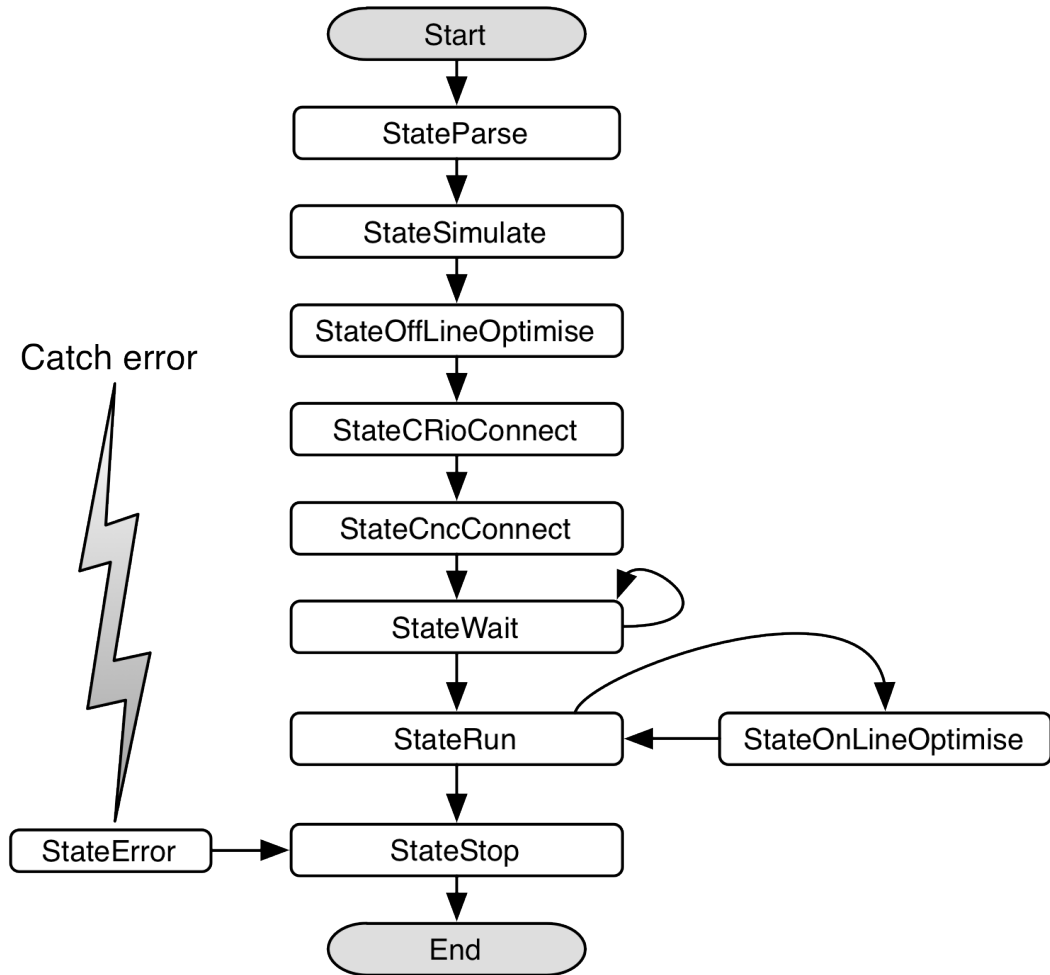


Figure 4.2: State machine of the EPC.

1. *StateParse*: the G-code is loaded, parsed, and the velocity and positions profiles are generated.
2. *StateSimulate*: the instantaneous MRR, cutting force, and SLD are calculated by computing the cut cross section along the toolpath.
3. *StateOffLineOptimise*: the OCP is solved offline.
4. *StateCRioConnect*: the UDP connection with the CRio is initialised.
5. *StateCncConnect*: the CORBA connection with the CNC is initialised.
6. *StateWait*: the state machine waits until the user starts the execution of the part program on the NC user pannel.
7. *StateRun*: the optimal controls are communicated to the NC, the sensors displayed in the working area are acquired, and the current status of the machine is reconstructed.
8. *StateOnlineOptimise*: with the updated model of the process the controls are re-optimised online.
9. *StateStop*: the EPC is disconnected from the CNC and CRio, and a log file with all the records of the process is stored.
10. *StateError*: errors risen during the process are caught and the system is shut down. This is a particularly important state since it increases the reliability of the system during critical situations.

In the next sections are described the functionalities and features of the simulator, OCP solver, and perception modules of the EPC.

4.1 Process simulator

The process is simulated at two levels: at first the tool path is reconstructed by analysing the programmed part program, then the cut is simulated by calculating the cut geometry and the dynamics of the cut.

4.1.1 Ruby Numerical Control

The numerical control implemented into the EPC is a simplified CNC software developed by the Authors (95). The numerical control software is written in the Ruby language¹ (and thus called RNC, *Ruby Numerical Control*). It supports a subset of the ISO G-Code language, and implements acceleration-limited velocity profiles, as well as three-axes linear and circular interpolation.

Linear (G01) and circular (G02 and G03) interpolation commands are directly implemented into the RNC software, while rapid positioning blocks (G00) and other non-machining instructions are not of interest for optimisation of spindle speed and feed rate, and are thus transparently passed through the parser (*no-op*).

The reference machine tool architecture adopted is that of a 3-axes milling machine tool with the kinematic chain depicted in Fig 3.2.

Generation of Feed Velocity Profiles

The *profiler* module cited in the previous section has the role of calculating a profile (or time-sequence) of tool tangent velocity (or instant feed rate) for each G01, G02, or G03 block, given the maximum available accelerations of the axes and the nominal feed rate specified in the G-Code. Once the feed velocity profile is known, it is used

¹<http://www.ruby-lang.org>

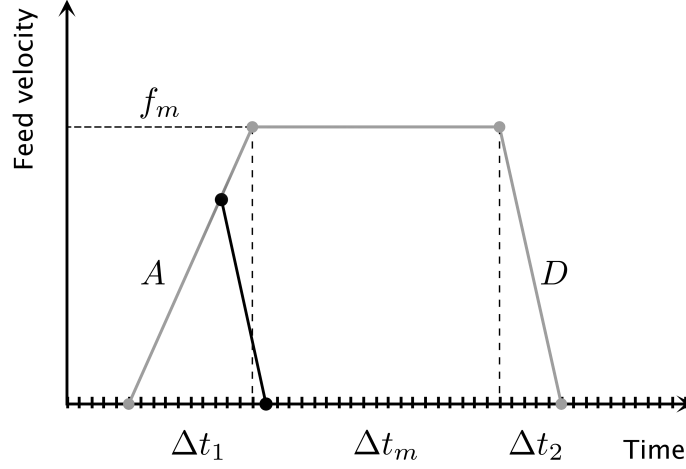


Figure 4.3: Trapezoidal (acceleration-limited) feed velocity profile. Short blocks can result in triangular profiles. Ticks on time axis represent the quantisation time t_q

to calculate the time history of the axes positions (operation that is usually called *interpolation* (96)).

There are numerous available approaches to the problem of feed velocity profiles generation, which usually try to provide profiles that minimise jerk and path following errors. The simplest model of profiler, though, assumes that the tool feed velocity goes to zero at the beginning and at the end of each block, and that the nominal feed rate f_m is reached within each machining block after a constant acceleration step and before a constant deceleration step. The resulting profile has the trapezoidal shape depicted in Fig. 4.3. In case of short blocks, the acceleration and deceleration steps could not allow to reach the nominal feed rate f_m , and in these cases the resulting feed velocity profile is triangular. It is worth noting that the resulting piecewise profile has discontinuous acceleration (and thus infinite jerk) on the profile corners. This is usually an undesirable condition, since high jerk values cause vibrations that can harm the surface finish and the tool itself. Also, the condition of null velocity at the nodes between blocks can result in an actual feed rate significantly *and needlessly* reduced with respect of the nominal one when a sequence of very short, almost collinear positioning blocks is

performed. Nevertheless, in order to evaluate the effectiveness of process optimisation by continuous variation of the feed rate (and spindle speed), neither the jerks nor the case of short collinear blocks are of concern.

To compute a trapezoidal velocity profile, the time duration of the two acceleration and deceleration steps, Δt_1 and Δt_2 , are estimated as:

$$\Delta t_1 = f_m/A \quad (4.1)$$

$$\Delta t_2 = -f_m/D \quad (4.2)$$

where A and D are the maximum acceleration and deceleration available, and f_m is the nominal feed rate. The integral of the velocity profile is equal to the block length L , which is defined by the G-Code:

$$\int_0^{\Delta t} f(\tau) d\tau = L \quad (4.3)$$

where $\Delta t = \Delta t_1 + \Delta t_m + \Delta t_2$. Consequently, the value of Δt_m is:

$$\Delta t_m = \frac{L}{f_m} - \frac{\Delta t_1}{2} - \frac{\Delta t_2}{2} \quad (4.4)$$

Since the axes drive controllers work on a discrete time basis, the trapezoidal velocity profiles are evaluated at discrete time steps $t_k = k t_q$ $k \in \mathbb{N}$, where t_q is the quantisation time, thus the block duration Δt is a multiple of the quantisation time. This quantisation is obtained by extending Δt_m up to its nearest upper quantised value, Δt_m^* , keeping constant Δt_1 and Δt_2 , and reducing f_m to a value f_m^* that satisfies Eq. 4.3. In

formulas:

$$\Delta t^* = \text{ceil}(\Delta t/t_q) t_q = \Delta t_1 + \Delta t_m^* + \Delta t_2 \quad (4.5)$$

$$\Delta t_m^* = \Delta t_m + (\Delta t^* - \Delta t) \quad (4.6)$$

$$L = \int_0^{\Delta t^*} f(\tau) d\tau = f_m^* (\Delta t_1/2 + \Delta t_m^* + \Delta t_2/2) \quad (4.7)$$

$$f_m^* = \frac{2L}{\Delta t_1 + \Delta t_2 + 2\Delta t_m^*} \quad (4.8)$$

$$0 < a = \frac{f_m^*}{\Delta t_1} < A \quad (4.9)$$

$$0 > d = -\frac{f_m^*}{\Delta t_2} > D \quad (4.10)$$

where a and d are reduced acceleration and deceleration as a consequence of the reduction of f_m down to f_m^* . Assuming that each profile starts at $t_0 = 0$, the resulting velocity profile is:

$$f(t) = \begin{cases} at & t < \Delta t_1 \\ f_m^* & \Delta t_1 < t < \Delta t_1 + \Delta t_m^* \\ f_m - d(t - \Delta t_1 - \Delta t_m^*) & t > \Delta t_1 + \Delta t_m^* \end{cases} \quad (4.11)$$

Note that for short tool movements, the accelerations A and D might not allow the tool to reach the desired feed rate f_m . In this case, which occurs when $\Delta t_m^* < 0$, the velocity profile is triangular. Its calculation is performed from analogous considerations made on the trapezoidal profiles, and it is omitted here for brevity.

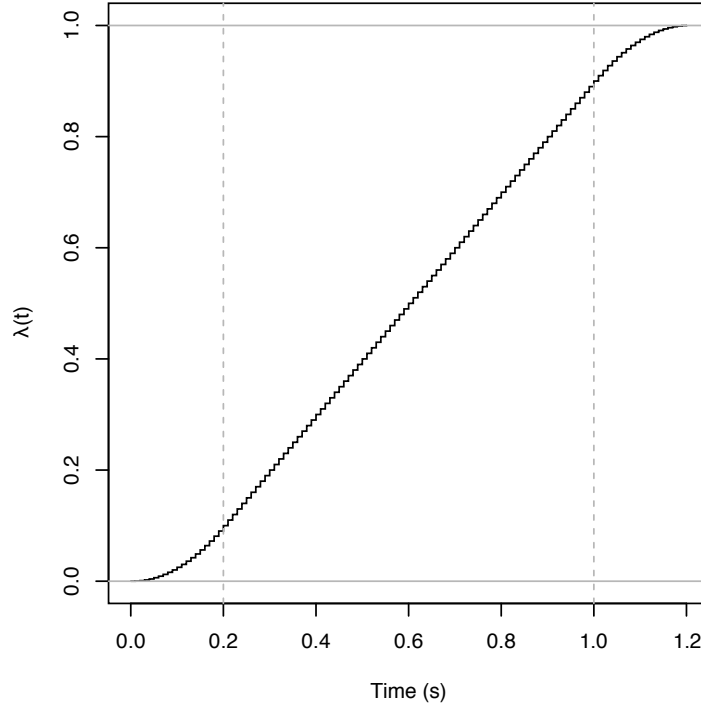


Figure 4.4: Function $\lambda(t)$ evaluated with a sampled time of 0.01 s, $\Delta t = 1.2$ s, and maximum accelerations $A = -D = 5$ m/s²; vertical dashed lines limit the constant feed rate zone

Interpolation of Axes Position

The velocity profile evaluated by the Eq. 4.11 is analytically integrated and normalised by the block length L , thus obtaining the λ function (Fig. 4.4):

$$\lambda(t) = \frac{1}{L} \int_0^t f(\tau) d\tau \quad (4.12)$$

where t is the time elapsed since the block start, and where it has to be noted that $\lambda(0) = 0$ and $\lambda(\Delta t^*) = 1$.

Let $\vec{s}_0 = (x_0, y_0, z_0)$ be the vector with the three components of the tool position when the current block starts, and $\vec{\Delta s} = (\Delta x, \Delta y, \Delta z)$ the vector of the total travel made

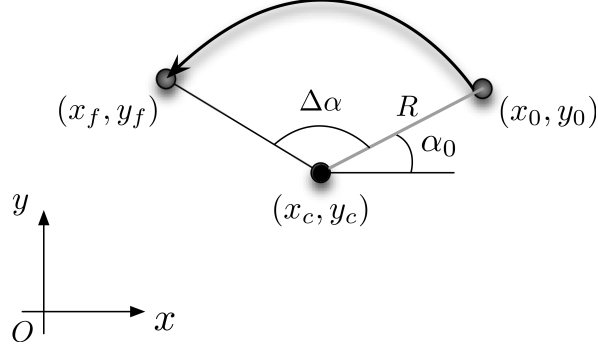


Figure 4.5: Representation of circular interpolation.

during the current block interpolation. The instantaneous tool position $\vec{s}(t_k)$ for a linear interpolation is evaluated as:

$$\vec{s}(t_k) = \vec{s}_0 + \vec{\Delta s} \lambda(t_k) \quad (4.13)$$

The RNC can also perform circular interpolations with rotation around the three cartesian axes. As instance, if the rotation is around Z at the k -th time instant the $z(t_k)$ position is evaluated with the corresponding Eq. 4.13, while $x(t_k)$ and $y(t_k)$ are:

$$\begin{aligned} x(t_k) &= x_c + R \cos(\alpha_0 + \Delta\alpha \lambda(t_k)) \\ y(t_k) &= y_c + R \sin(\alpha_0 + \Delta\alpha \lambda(t_k)) \end{aligned} \quad (4.14)$$

where α_0 is the phase of the vector joining the arc centre C and the start point O of the block, $\Delta\alpha$ is the angle included by the interpolated arc, and R is its radius (see Fig. 4.5).

4.1.2 Cut simulator

A cut simulator is included into the EPC to calculate the Material Removal Rate (MRR), cutting force, and Stability Lobes Diagrams (SLD) along the curvilinear abscissa of the tool path. The simulator has been implemented in C/C++ by the group of Mechatronic Engineering of the University of Trento, and it is based on the Spatial ACIS© geometric modelling kernel. The main advantage of the ACIS library is its capability to compute intersections between complex workpiece and tools geometries moving along complicate tool paths. Tool and the raw workpiece are modelled as 3D solids and are reinterpreted as a series of canonical geometries (circles, lines, planes), which are parameterised into polynomials. Mathematically, the position of these geometries in the 3D space is represented by a non linear system of algebraic equations, whose solution gives the intersections.

The tool path is discretised into a finite number of steps, at which it is calculated the interference, i.e. the engagement, of the mill into their workpiece. The engagement is described by the cut cross section, as well as by the entry and exit angles of the cutters. Once the cut material has been calculated, the geometry of the workpiece is updated, the mill is placed at the next position along the curvilinear abscissa, and the computation of the intersections is repeated.

The information about the engagement is used to calculate the cutting force vector, and the stability lobes diagram. Both are given by the methodology proposed by Altintas and Budak (53) (see Appendix B).

4.2 OCP solver

The scheme of the OCP solver is in Fig. 4.6. The OCP is formulated in *XOptima*©, an optimisation library developed by the Mechatronic Engineering group of the University

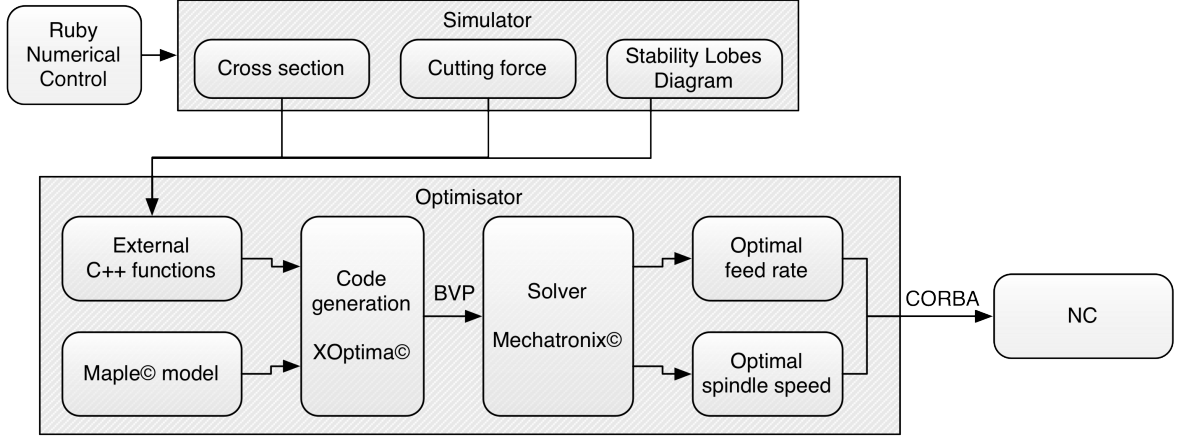


Figure 4.6: Architecture of the OCP code generator and solver.

of Trento. XOptima is installed as an add-on package for Maple© software, which is used to develop the equations of the model of the dynamical system. Target function, DAE constraints, and initial conditions are processed and formulated as a Boundary Value Problem (BVP); the OCP is then automatically converted into a C++ code by the Maple CodeGeneration package. Thanks to the followed approach, it is possible to define external C++ functions and include them into the BVP formulation, thus easing the integration of complex models into the OCP.

The BVP is compiled and linked to the OCP solver library *Mechatronics©*, implemented in C++ by the Mechatronic Engineering group of the University of Trento. Once the OCP is solved, the controls are communicated to the NC through the CORBA infrastructure.

4.3 Learning layer

The online optimisation is performed once the instantaneous status of the process has been reconstructed. Such information is given by the set of data recorded from NC,

and sensors displayed in the working volume. In order to have a robust and reliable state reconstruction, it is necessary to fuse all the collected measurements.

The parameter with respect to which the model of the process is more sensitive, is the specific energy of the cut material u_s . It is defined as the energy required to cut 1 mm³ of material, and it is calculated from the Eq. 2.22:

$$u_s = \frac{P_c}{MRR} \quad (4.15)$$

Particular attention should be given to the adopted unit of measurements since the material removal rate is expressed in mm³/min, while u_s is expressed in J/mm³. The cutting power can be estimated online, by measuring the spindle speed and the cutting torque directly from the NC. In this case the specific energy becomes:

$$u_s = \frac{\omega T_c}{MRR} \quad (4.16)$$

The cutting torque is measured from the feedback loop on the spindle speed. As a consequence, it tends to be noisy and change significantly when the cutting force changes abruptly. A filter is used to improve the estimation of the instantaneous u_s . The first step consists in individuating the sections of the tool path where the measurements and the process are stable. The estimation of u_s is considered valid when three indicators reach a *true* value:

1. D_0 is an index defined on the measurement percentage error $\epsilon\%$ of u_s . It avoids to learn values of the specific energy that are too different with respect to those reported in literature and used for the offline optimisation. At the k -th position

of the mill, D_0 is calculated as:

$$D_{0,k} = \begin{cases} 1 & \text{if } \epsilon_{\%,k} \leq 25\% \\ 0 & \text{otherwise} \end{cases} \quad (4.17)$$

$$\epsilon_{\%,k} = \frac{|\bar{u}_{s,k} - u_{sn}|}{u_{sn}} \quad (4.18)$$

$$\bar{u}_{s,k} = \frac{\sum_{i=0}^N u_{s,k-1}}{N} \quad (4.19)$$

where u_{sn} is the nominal specific energy, $u_{s,k}$ is the measured one, and $\bar{u}_{s,k}$ is the average of the last N measurements. The latter takes into account not only the most recent measurement, but also its records. The number of samples to be considered, i.e. the amount of time to which look backward in the history, is $N = 15$. It corresponds to 3 s of cut, since the sampling frequency is 5 Hz. This amount of time has been chosen as a compromise between the time required to ensure that the cut reaches a stable condition, but also to avoid to include an excessive delay on the state estimation procedure.

2. D_1 estimates the variation of u_s along the curvilinear abscissa, such that local modifications of the cutting torque due to noise of measurement or transitory events —e.g. when the mill enters into the workpiece— do not concur to estimate the specific energy. On contrary, variations of the material properties —e.g. due to local thermal treatments— modify the cutting torque with slower dynamics, and this event is captured and learnt by the EPC. D_1 is true whether the specific u_s does not double its value in 1 mm of travelled tool path:

$$D_{1,k} = \begin{cases} 1 & \text{if } \left| \frac{\Delta u_s}{\Delta \zeta} \right|_k \leq u_s \\ 0 & \text{otherwise} \end{cases} \quad (4.20)$$

$$\left. \frac{\Delta u_s}{\Delta \zeta} \right|_k = \frac{u_{s,k} - u_{s,k-1}}{\zeta_{s,k} - \zeta_{s,k-1}} \quad (4.21)$$

3. D_2 estimates the curvature (i.e. the second derivative) of the specific energy with respect to the curvilinear abscissa. This quantity is used to evaluate the trend of u_s . When the second derivative is large, the jerks on the spindle tend to be high, therefore in such instants the measurement is discarded. D_2 is defined as:

$$D_{2,k} = \begin{cases} 1 & \text{if } \left. \frac{\Delta^2 u_s}{\Delta \zeta^2} \right|_k \leq u_s \\ 0 & \text{otherwise} \end{cases} \quad (4.22)$$

The threshold u_s is set by assuming that the measured cutting torque is stable if the specific energy increases less than twice its value in 1 mm^2 of cut cross section. The second derivative is calculated by the backward difference quotient:

$$\left. \frac{\Delta^2 u_s}{\Delta \zeta^2} \right|_k = \frac{1}{\Delta \zeta} \left(\left. \frac{\Delta u_s}{\Delta \zeta} \right|_k - \left. \frac{\Delta u_s}{\Delta \zeta} \right|_{k-1} \right) \quad (4.23)$$

When D_0 , D_1 , and D_2 are true (i.e. 1) it is available a measurement of u_s enough robust to be integrated into the sensor fusion layer. According to the considerations here made, when the process is stable it is assumed that the noise of measurement: has null mean, has a power spectral density (PSD) uniformly distributed along the whole frequency band considered, is normally probability distribution, and the samples are uncorrelated along the time. It is so assumed that the measurements are affected by white noise. These assumptions are needed in order to utilise the state estimators commonly used in literature.

4.3.1 Sensor Fusion

The most famous algorithms used for the sensor fusion are the *Kalman Filter* (KF) (10) and the *Particle Filter* (PF) (11). Kalman filter is used in this study because it is an efficient algorithm and it is easy to implement in the scalar case. KF requires that the model of the process is linear with respect to the state variables. Such condition is not satisfied by the model in Eq. 4.16, where the state is constituted by the cutting torque, MRR, and spindle speed. A common solution to this problem is to linearise locally the non linear model, or to find a proper transformation of the state with respect to which the model becomes linear. These approaches are called Extended Kalman Filter (EKF) and Unscented Kalman Filter respectively (97; 98).

In the EPC it is used a modified version of KF, where the prediction phase is substituted by the value of specific energy known from the literature. By doing so it is avoided that the learnt u_s diverges excessively from its reference value. During a cut indeed, the eventuality in which u_s changes significantly with respect to its nominal value might be due to an improper setup of the OCP, therefore it represents an extremal condition which must be handled not by correcting the estimated u_s , but rather by redesigning the process optimisation.

The update of the specific energy at the k -th instant is calculated as:

$$\tilde{u}_{s,k} = \frac{u_{sn} \sigma_m^2 + u_{s,k} \sigma_n^2}{\sigma_m^2 + \sigma_n^2} \quad (4.24)$$

where σ_m^2 is the variance of the nominal specific energy and it is obtained from the literature, assuming that the data reported in the manuals are affected by an uncertainty of 50 %. The variance of the measurement is $\sigma_n^2 = 1.3 \cdot 10^{-3} \text{ J}^2/\text{mm}^6$, and it is estimated by propagating the uncertainty associated to T_c and ω across the Eq. 4.16. The latter have been estimated by cutting a fixed cross section of workpiece with a constant spindle speed and feed rate, and their values are $\sigma_{T_c} = 2.5 \text{ Nm}$ and $\sigma_\omega = 0.72 \text{ rad/s}$.

4.3.2 Validation

The learning layer has been validated for three different face milling processes: with full immersion and constant cross section (Fig. 4.7), with increasing cutting width (Fig. 4.8), and with a complicate path arbitrarily defined (Fig. 4.9). The cut material is the AA6082-T6.

For the three case studies are shown the plots of the nominal, learnt and measured specific energies. The former is equal to 0.7 J/mm^3 . The measured specific energy is represented by three colours: red for the sections of the tool path in which the mill is not cutting, orange for the parts in which the mill is cutting but the torque is excessively noisy, and green for the instants when D_0 is true. The same colour scheme is applied for the plots of D_1 and D_2 . At the bottom it is shown the cut cross section calculated by the process simulator, and it is green when the three index are true, that is when the EPC is actually learning the process.

From the results obtained it is observed that u_s is adjusted when the measurements are stable. The estimated specific energy is a linear combination of the measured and nominal ones, suitably weighted by the confidence with which those values are known. The proposed approach is robust even for highly noising measurements such as the ones in Fig. 4.8.

The results here obtained allow to maintain the models of the process constantly updated, therefore this learning layer can be exploited to the online adaptive control of the cut.

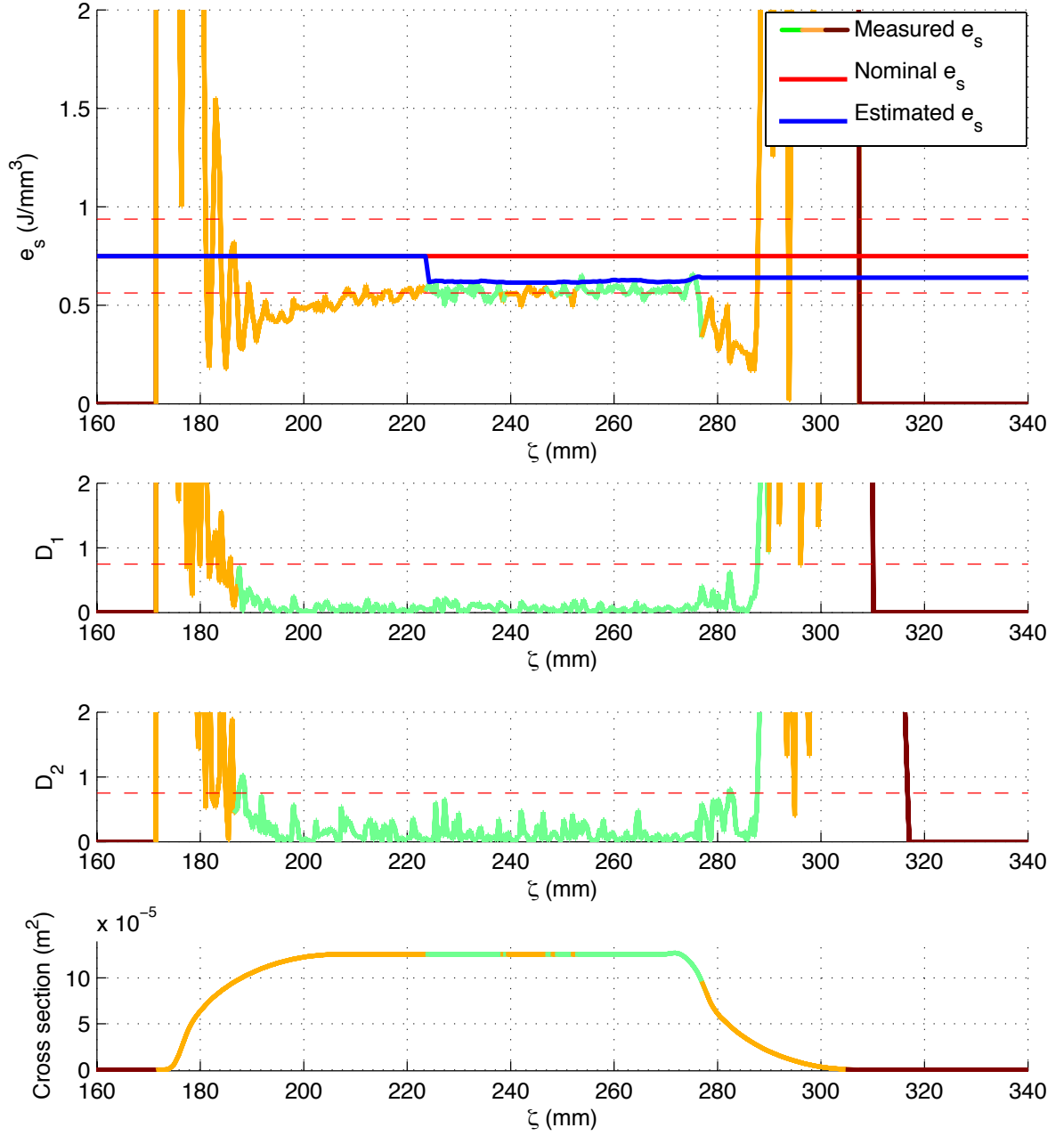


Figure 4.7: Validation of the learning layer for the full immersion face milling.

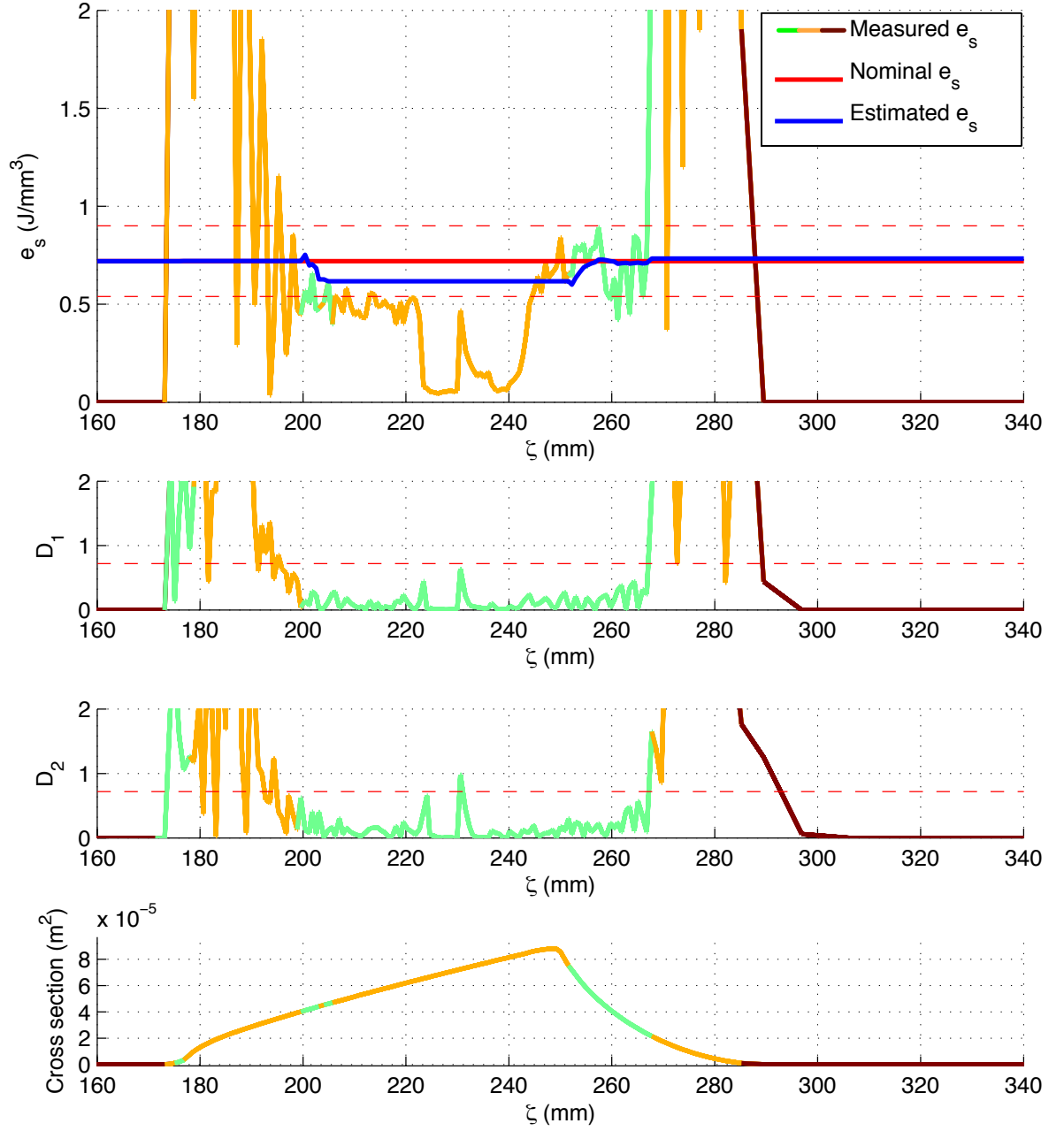


Figure 4.8: Validation of the learning layer for the face milling with increasing cutting width.

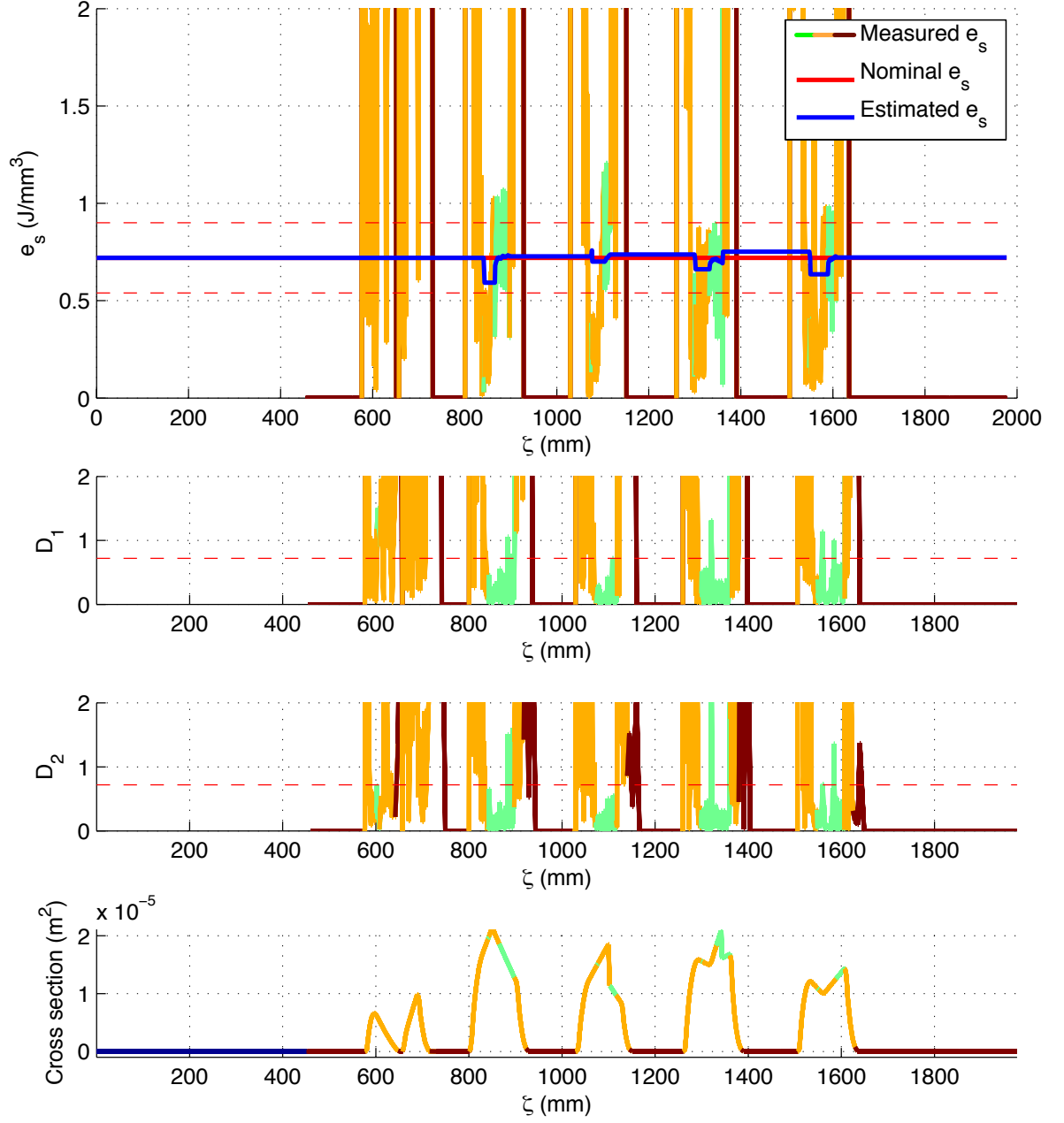


Figure 4.9: Validation of the learning layer for the face milling with complicate tool path.

Chapter 5

Communication Layers

In this chapter it is described interface developed for the communication between the EPC and the NC Fidia CN20. The architecture of the software is outlined and its performances are reported. A bi-directional protocol is used for the communication between EPC and NC, thus allowing the controller to read the status from the machine and to copy the optimal feed rate and spindle speed. As it will be shown in the next sections, the most time consuming operation is the communication of the optimal feed rate and spindle speed from the EPC to the user panel. According to the manufacturer of the NC Fidia, the mean time required to communicate one parameter to the NC is approximately 10 ms. Since two parameters are communicated at each optimisation loop, at least 20 ms for this operation are required. Moreover, it has to be considered that the EPC should work at more than 5 Hz. During this time, several operations need to be performed: the acquisition of the NC data, the measurement of the forces and accelerometer data from the CompactRio, the reconstruction of the whole status of the system, the online calculation of the optimal process parameters, and the communication of these to the NC. It is essential therefore to have an estimation of the latencies and task execution times associated to the communication.

In the next sections are described the tests performed to characterise the latency associated the communication between EPC and NC. The communication software is written in C++ and in Ruby, whilst the interface between EPC and NC exploits CORBA.

5.1 Common Object Request Broker Architecture

CORBA has been developed by the Object Management Group, and it is an independent architecture and infrastructure that allows applications to communicate across the network (99; 100; 101; 102). Thanks to this framework applications can communicate through the Internet Inter-ORB Protocol (IIOP), that is independent from the programming language and hardware in which they have been implemented.

CORBA is constituted by three elements:

1. Client: requires the services;
2. Server: provides the services;
3. ORB: manages the information exchange between client and server.

In this study the client is run in the EPC, while the server is run in the user panel that is interfaced with the NC. The software architecture of the optimisation system is schematised in Fig. 5.1. All the services provided by server and interface are defined in the Interface Description Language (IDL). Thanks to the flexibility of CORBA, it is possible to instantiate clients that require services even to multiple servers. The service requests raised from clients are intercepted and driven to the servers by the *skeleton*. When a service request reaches the servers, the IDL is used to decode the received parameters. The server utilises the same IDL to encode the outputs of the service, to send them back to the client, and to communicate to it how to decode the data when

they are received. By using the IDL the client invoke *remote* methods as if they were functions of a *local* instance of the server object, though the client is actually calling a *stub* that acts as a proxy. The stub is a local simulation of a remote object and it is interfaced with the IDL.

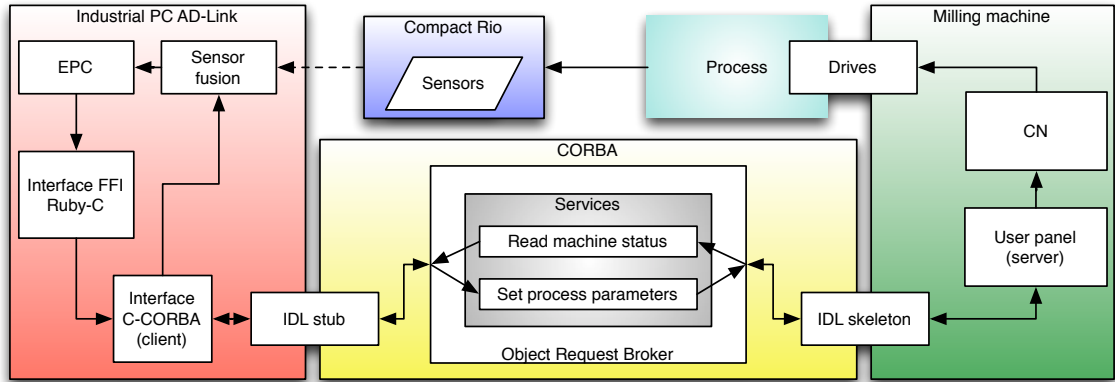


Figure 5.1: Architecture of the software developed for the communication between EPC and NC.

5.2 Interface Ruby - CORBA

The Ruby-CORBA interface is defined by two layers: the first layer allows the EPC to instantiate clients and to require services from the server, the second layer is the interface with CORBA, Fig. 5.2. The EPC is implemented in Ruby, while CORBA services are accessible via C++ classes. With this design of the communication layer it is possible to combine the advantages of Ruby and C/C++ thus obtaining an efficient software. Ruby is an interpreted language, highly flexible, and with a simple syntax. This language is ideal for implementing the whole architecture of the EPC. On contrary, C/C++ is a compiled language so it is particularly suitable to perform complex and time-consuming tasks like the handling of the complex CORBA architecture. It

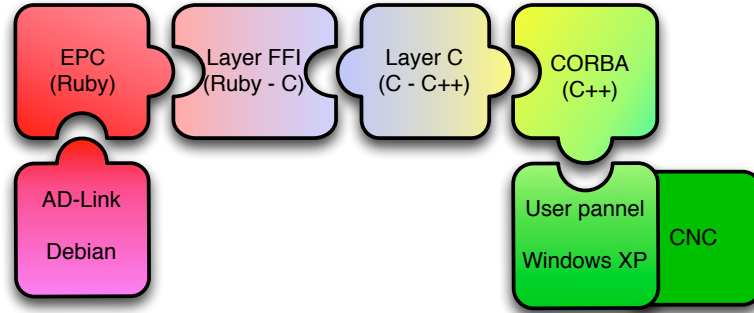


Figure 5.2: Communication interface and layer between EPC and NC.

is not possible to wrap C++ functions in Ruby due to lack of an Application Binary Interface (ABI) standard that makes the C++ source code independent from the compiler used. Nevertheless, the pure C code has the compilation standard *ANSI X3J11* —ANSI C—, and *ISOIEC JTC1SC22 WG14* —ISO C—. For Ruby is available the FFI library to wrap C and Ruby code ¹.

In Fig. 5.3 are outlined the modules and functions implemented in the communication layer between EPC and user panel. The EPC includes the Interface Ruby-CORBA (IRC) module containing the classes and the FFI functions necessary to communicate with C/C++ layer. The *ConnectionParameters* class allows the EPC to access to the *ConnectionPar* C structure, where there are stored the parameters used to configure the connection with the NC (IP-address, port number, and NC tag-name). The *MachineStatus* class is wrapped to the *Status* C structure, containing the status acquired from the NC.

The *Connection* class has all the methods necessary to manage the connection, set or override the process parameters, and to acquire the status of the machine (whose fields are listed in Appendix F.2). The most important FFI functions implemented are, and are grouped in three structures:

¹<https://github.com/ffi/ffi>

1. *connect_IRC*: initialises the connection and the Status structure;
2. *disconnect_IRC*: closes the connection and de-allocates the memory;
3. *set_parameters_IRC*: sets the feed rate and the spindle speed to a specific value.
4. *get_status_IRC*: acquires the instantaneous state of the machine.

The FFI methods are wrapped to C/C++ functions that access to the CORBA client. The architecture here presented allows a single client to require services from multiple servers, once a *ConnectionPar* structure has been defined for each communication channel. Note in Fig. 5.3 that the function *get_status_IRC* has a bidirectional arrow because it can be called by the EPC and returns the current status of the machine.

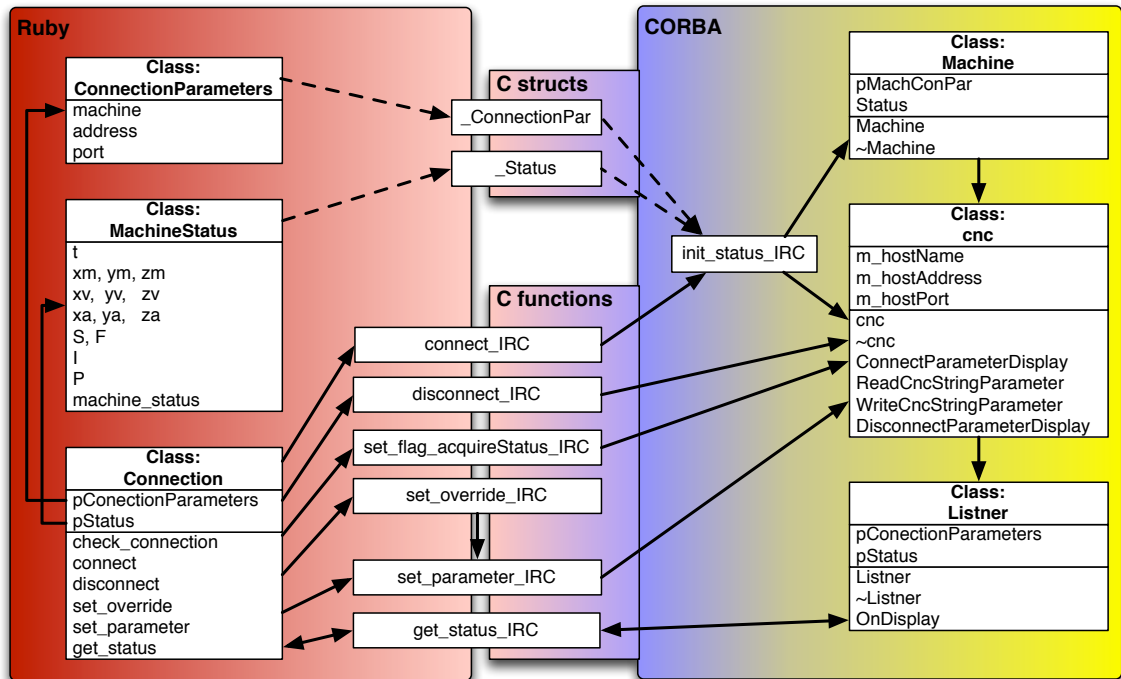


Figure 5.3: Classes architecture of the communication layer between EPC and NC.

On the CORBA side of the architecture, three C++ classes are defined: *Machine*, *cnc*, and *Listener*. The former is used to store the information about the connection, the second one initialises the CORBA connection, while the latter acquires the status of the machine by requiring the *OnDisplay* service to the CORBA server. This is done by utilising a parallel thread that is killed by the NC when it registers a modification of one or more of the listened parameters. The parameters and sub parameters that are monitored by the instance of the Listener are summarised in Appendix F:

1. *M.XAC*: returns the positions of the axes in the active workpiece frame.
2. *MIC_BLK_F_INFO*: retrieves the information about active block and executed file.
3. *MIC_INST_DATA*: collects the remaining information necessary to define the status of the system.

The NC measures the status of the machine at a higher frequency (1000 Hz) than the one used to communicate them to the EPC (≈ 500 Hz). Therefore, for some outputs, the NC measures the data as fast as possible and communicates to the user panel their statistical descriptors at a lower frequency (100 Hz). The chosen descriptors are mean, minimum, and maximum values, that are calculated between two consecutive data transmissions.

5.3 Characterisation and validation

Tests are performed to characterise and validate the communication layer between EPC and CORBA. TET and latencies are measured for both directions of the EPC-NC communication.

5.3.1 Task Execution Time

The communication between EPC and NC has to be fast, in order to have a controller that runs at 5 Hz. In this study fast operations are those accomplished in less than 100 ms. During the TET tests it is ensured that only the essential processes are active on the EPC and NC.

The tests carried out to validate the communication interface are performed by acquiring the state of the machine both through the EPC and the internal oscilloscope in the NC. The data acquisition is triggered by the NC (i.e. its operation mode), thus minimising the acquisition of non significant information.

It is possible to consider a null TET for the communication between EPC and Compact RIO since it is made through UDP that is a light and fast protocol. Moreover, the Compact RIO has built in a FPGA chip that ensures hard real time performances (sampling frequency >50 kHz). On contrary, the communication between EPC and NC is more complex due to CORBA and to the control panel that acts as a proxy to the NC.

Analysis of TETs are carried out by running the program List. 6.1 and measuring the time required to perform three different operations:

1. Writing on the user panel new values of feed rate and spindle speed.
2. Reading the machine status from the user panel.
3. Reading the status, and subsequently send the new feed rate and spindle speed to the user panel.

The measurements are made on the EPC side, and the results are reported in Tab. 5.1 and Fig. 5.4. It is possible to observe that the fastest operation is the reading of the

machine status, with an average TET of 0.01755 ms. To write two parameters on the user panel it takes 32.300 ms, therefore one parameter is updated in 16.150 ms that is in agreement with the TET measured by the NC manufacturer (Tab. 5.2). Outliers in Fig. 5.4 are due to stochastic delays of the non-realtime operating systems running on the EPC (*Linux Debian*) and on the user panel (*Windows XP*).

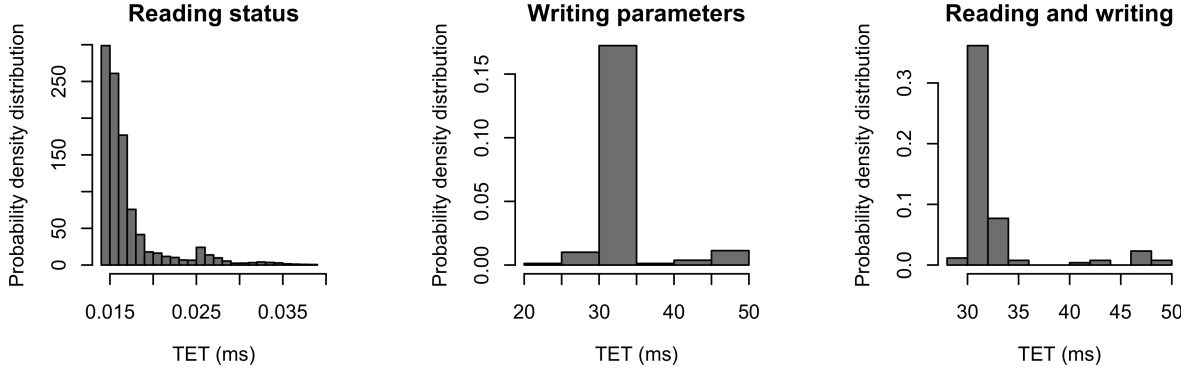


Figure 5.4: Histograms of the probability density distributions of TETs measured when reading the status of the machine (left), writing two parameters on the NC (centre), and performing both operations (right).

Table 5.1: Measured TETs for different operations, with 95% of confidence.

Operation	Mean TET (ms)	Standard deviation (ms)	95 % confidence interval (ms)
<i>Read the status</i>	0.01755	0.00388	[0.01752, 0.01757]
<i>Write 2 parameters</i>	32.300	4.133	[31.653, 32.947]
<i>Read - write</i>	32.545	4.374	[31.786, 33.304]

Table 5.2: TETs required to write one parameter on the NC according to the NC manufacturer FIDIA.

Parameter	Value (ms)
<i>Minimum latency</i>	3.291
<i>Maximum latency</i>	10.019
<i>Median latency</i>	9.787
<i>1-st percentile</i>	4.275
<i>99-th percentile</i>	10.018
<i>99.9-th percentile</i>	10.019
<i>Standard deviation</i>	1.028

5.3.2 Validation of the acquisitions

The data acquired with CORBA are validated by comparing them with measurements collected from the internal oscilloscope on the user panel. The programmed tool path is a spiral with a radius of 25 mm. The experiment is repeated twice, thus acquiring the status of the machine through both the EPC and oscilloscope. In Fig. 5.5 are compared measured positions and feed rates for the three cartesian axes X , Y , Z .

The acquisitions made through CORBA are in agreement with the collected data from the oscilloscope. Moreover, the former are offset in advance of approximately 170 ms with respect to the latter, meaning that the user panel manages the CORBA communication process with a higher priority than the oscilloscope process. It is a desired feature, and it is necessary in order to have a reliable communication with the EPC, especially when multiple operations are executed on the user panel.

5.3.3 Latencies and sampling frequencies

To have a robust optimal control, it is necessary to characterise the time associated to the EPC-user panel communication, the latencies associated to it, as well as the

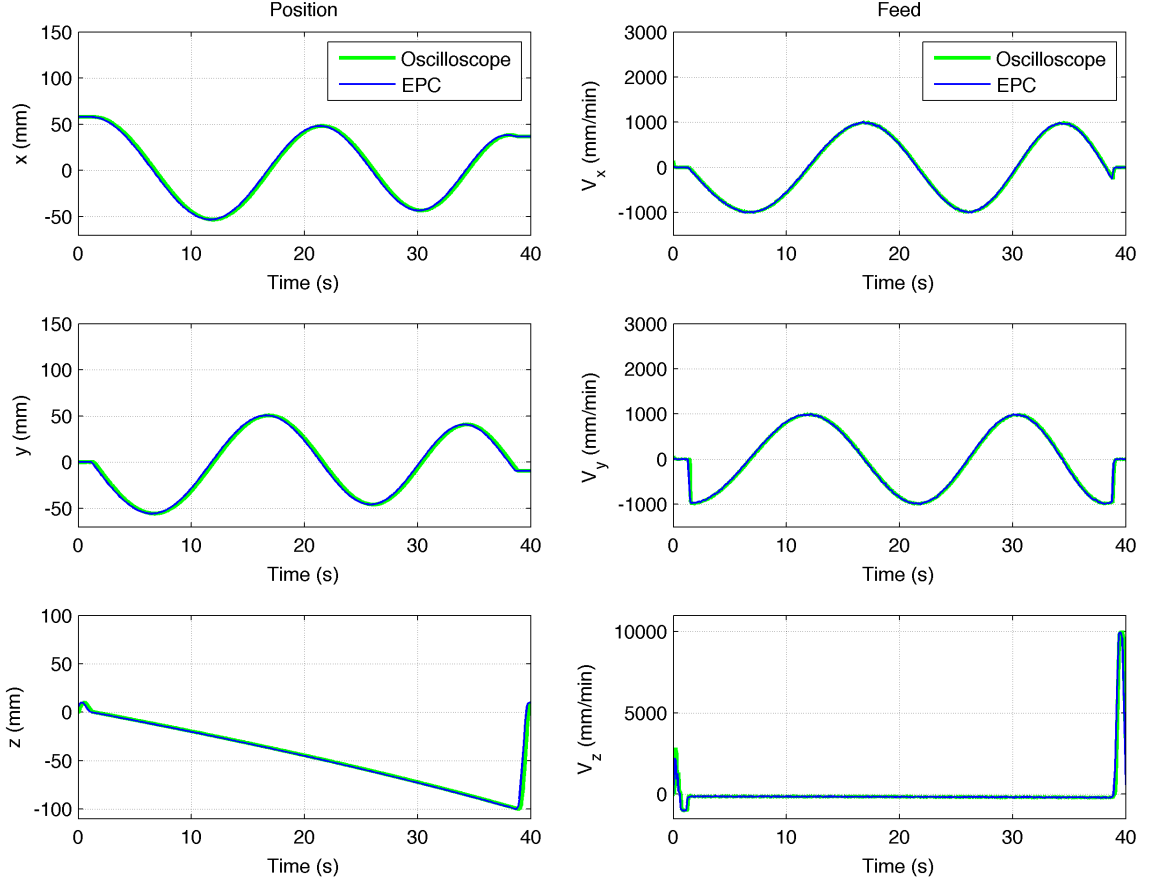


Figure 5.5: Comparison of the measurements of positions and velocities for the three axes, made with the EPC and with the oscilloscope internal to the user panel of the machine.

sampling frequencies. It means to estimate the time Δt_{EPC-NC} required to send a parameter from the EPC to the NC, and the time Δt_{NC-ME} elapsed since the parameter is updated on the NC and it is measured back by the EPC. The information exchanged is the feed rate F and the spindle speed S .

Δt_{EPC-NC} is the sum of TET_{EPC-UP} necessary to communicate one parameter to the user panel, and TET_{UP-NC} associated to the communication between the user panel and NC:

$$\Delta t_{EPC-NC} = TET_{EPC-UP} + TET_{UP-NC} \quad (5.1)$$

Δt_{NC-ME} is obtained by the sum between the actuation time Δt_a necessary to accelerate the axis (or the spindle) to the nominal F (or S), and TET_{NC-EPC} required to read the status of the machine that can be neglected as exposed in Sec. 5.3.1:

$$\Delta t_{NC-ME} = \Delta t_a + TET_{NC-EPC} \quad (5.2)$$

The total latency Δt_L is thus:

$$\Delta t_L = \Delta t_{EPC-NC} + \Delta t_{NC-ME} \quad (5.3)$$

The tests are performed by moving only the X axis, while linearly modifying the feed rate from 0 mm/min to the maximum value 10000 mm/min . The EPC sends to the user panel a periodical triangular feed profile with period 1 s . The measurements for one period are reported in Fig. 5.6. *Sent* is the time at which the nominal F is sent by the EPC, *set* is instant when F is received by user panel, and *measured* is the time at which the feed rate reaches physically the nominal value.

This test allows to identify the minimum sampling frequency that has to be set on the NC. From preliminary tests it is observed that the set triangular feed profile (dashed line) has a quantised trend (Fig. 5.6 on the left). This is due to the fact that the communication between the EPC and the user panel is fast ($\approx 30 \text{ Hz}$), while the sampling frequency between user panel and NC is slower ($\approx 10 \text{ Hz}$). The EPC reads for several times the same data from the NC, thus resulting in a quantised acquisition. Note that the same trend is observed for the measured feed rate (dot-dashed line), but here it is not appreciable because the data are pre-filtered by eliminating the duplicate values in order allow the calculation of the acceleration for further analysis. The data quantisation problem is addressed by increasing the user panel-NC sampling frequency to 100 Hz , thus obtaining smoother data, see Fig. 5.6 on the right.

The latencies are measured by time-shifting and superimposing the *sent* and the *mea-*

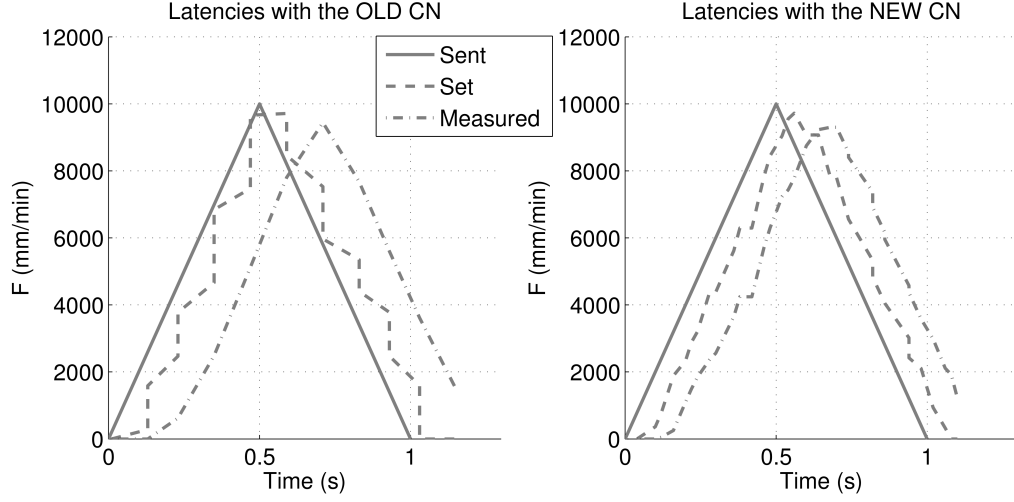


Figure 5.6: Feed rate profiles used to estimate the latencies measured when acquiring from the NC at ≈ 10 Hz (left), and ≈ 100 Hz (right).

Table 5.3: Measured latencies.

Operation	Sampling freq. 30 Hz	Sampling freq. 100 Hz
Δt_{EPC-NC}	60 ms	60 ms
Δt_{NC-ME}	150 ms	115 ms
Δt_L	210 ms	180 ms

sured curve to the *set* lines. The time shifts are calculated through a manual bisection method, by minimising the residual sum of squares, and the results are summarised in Tab. 5.3. The total latency is 180 ms. This results means that during the online process control the EPC will have to anticipate the sent controls in order to be in-phase along the tool path with the controls profile calculated in the OCP.

The total latency is given by the contribute of different sources of delay:

1. Communication of the new feed from the EPC to the user panel. It takes 16.15 ms for each sent parameter.
2. Communication of the new feed from the user panel to the NC. According to

Eq. 5.1 it is equal to $TET_{UP-NC} = 43.85$ ms.

3. Actuation dynamics, that is estimated equal to $\Delta t_a = 115$ ms from Eq. 5.2. During this period, the NC modifies the nominal feed rate in order to take into account the maximum linear acceleration, to respect the quantisation time of the NC, and to limit jerks. This regularising effect is applied not only when computing the jerk-limited velocity profiles of the tool path, but also when the feed rate is instantaneously changed.
4. Measurement of the machine state through the EPC. As seen in the previous chapter this operation can be considered instantaneous but, due to the non-realtime nature of the operating systems running on the EPC and on the user panel, the acquisition might be affected by stochastic delays.

Chapter 6

Machine Identification

The goodness and robustness of the controls calculated by the Optimal Control Problem depend on the accuracy of the models used to describe the milling process, machine kinematics, and machine dynamics. The more accurate the models of the system are, the better is the optimal solution. In this section are described the tests carried out to characterise the machine dynamics. These tests are focussed on the identification of velocity profiles, maximum feed rate, and accelerations of axes and spindle of the machine *Alesamonti MB-63*. The dynamics of the drives are studied by changing the feed rate instantaneously (stair input), periodically (sweep input), and continuously (ramp input). The spindle is studied with tests aimed at identifying the free run performances and losses.

6.1 Dynamics of axes

6.1.1 Maximum feed rates and accelerations

The maximum feed rate and acceleration for each axis are studied by setting the feed rate from zero to the maximum value 10000 mm/min with step and ramp inputs. The tool path designed for the step input test is reported in List. 6.1.

```

1 N1 (Fidia S.p.A. ISOGRAPH v. 8.2.3 C1 10/12/12 18:58:42)
  N2 (Compilation of File: Y:\identification_rapid_max)
3 N3 G00 X0.000 Y0.000 Z0.000 B0.000 W0.000 F10000 S1500
  N4 X600.000 M03
5 N5 Z600.000 M05
  N6 B90.000
7 N7 W-250.000
  N8 Y260.000
9 N9 X0.000 Y0.000 Z0.000 B0.000 W0.00

```

Listing 6.1: Part program executed during the characterisation of the axes.

The axis are moved one at time with the sequence: X , Z , B , W , and Y . Each axis is moved though a path long enough to reach the nominal feed rate, in order to have trapezoidal velocity profiles. The last block brings back all the drives to the origin with a rapid movement, thus allowing to verify that the maximum feed rate and acceleration achieved during the interpolated movement are limited by the slowest drive. The movements are repeated five times, and they are done in air, in order to minimise the risk for the machine and the worker man.

The accelerations are calculated from the measured velocities through a finite difference scheme, after filtering the data with a moving mean. The axes are moved with two approaches: with linear interpolated $G01$ movements where the feed rate is set at its

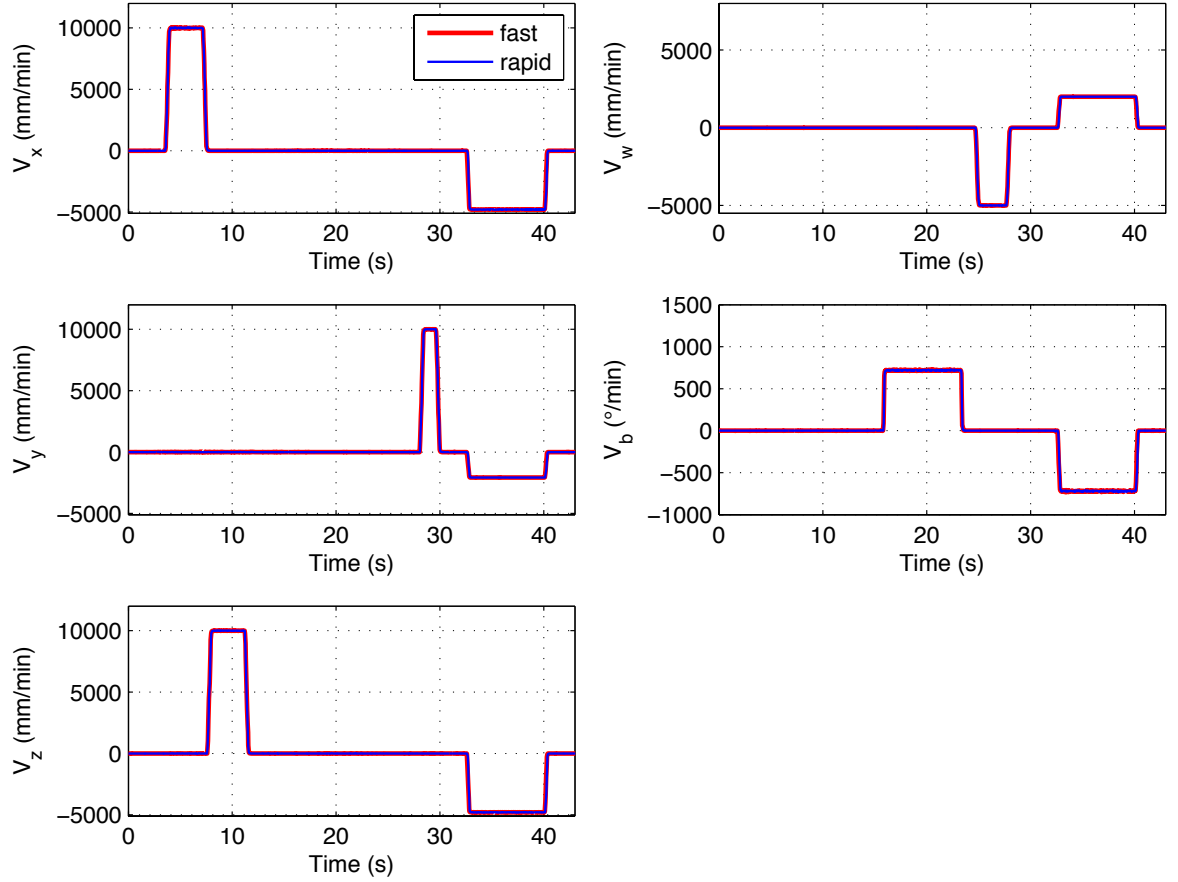


Figure 6.1: Feed rates of the axes for the test performed giving as input: feed steps from 0 mm/min to 10000 mm/min (red), and rapid movements (blue).

maximum theoretical value (10000 mm/min), and with rapid *G00* movements. This is done to verify if the latter are interpolated too.

The measured velocity profiles and calculated accelerations are shown in Fig. 6.1 and Fig. 6.2 respectively. Note that in the two figures, between 30 s and 41 s, velocities and accelerations are lower. This is due to the fact that in the last block the dynamic of the system is actually limited by the slowest drive —*B*— which rotates the working table. Tab. 6.1 summarises the measured maximum velocities and accelerations for the five tests. The former are in agreement with the data reported on the data sheet of

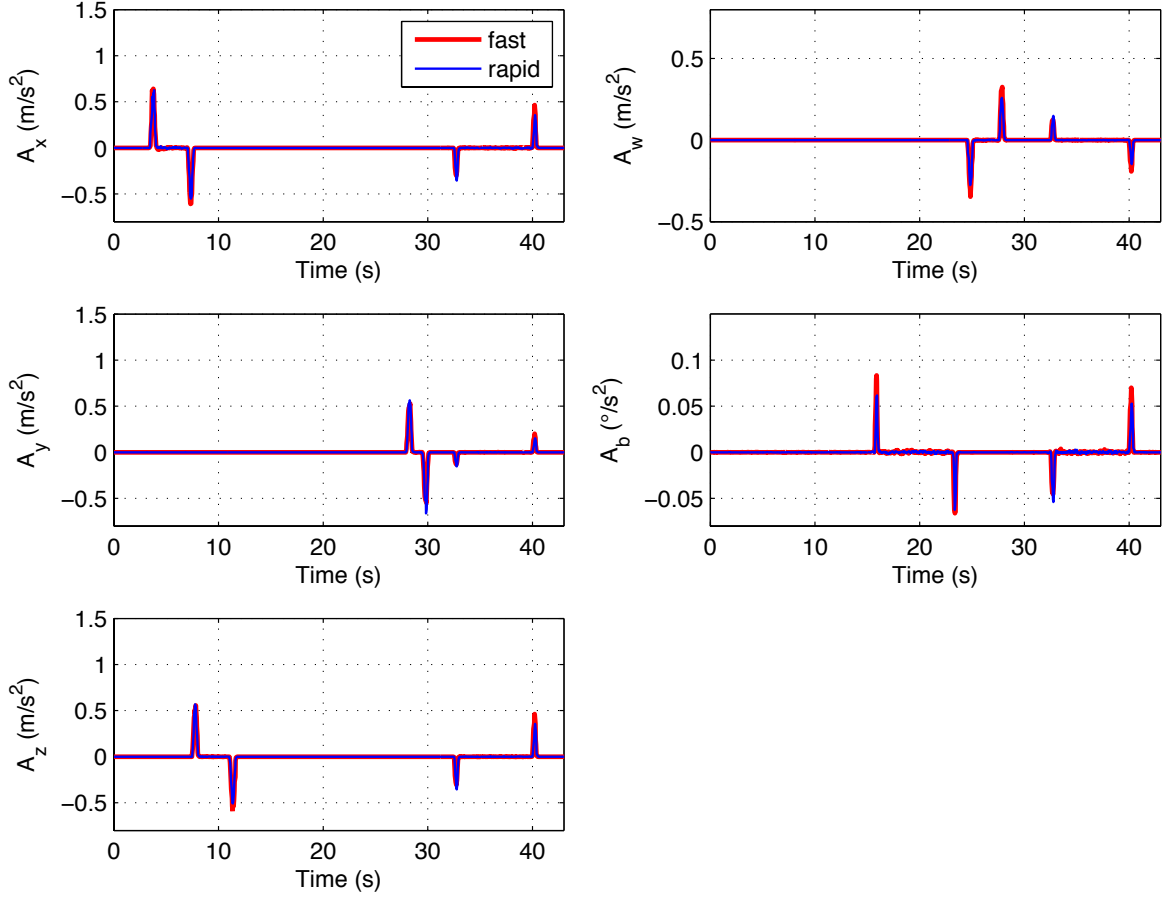


Figure 6.2: Accelerations of the axes for the test performed giving as input: feed steps from 0 mm/min to 10000 mm/min (red), and rapid movements (blue).

Table 6.1: Maximum velocities acceleration and deceleration of the axes.

Axis	Velocity	Acceleration	Deceleration
X	$9997.4 \pm 18.1 \text{ mm/min}$	$0.580 \pm 0.033 \text{ m/s}^2$	$-0.572 \pm 0.032 \text{ m/s}^2$
Y	$9999.6 \pm 5.8 \text{ mm/min}$	$0.522 \pm 0.015 \text{ m/s}^2$	$-0.557 \pm 0.025 \text{ m/s}^2$
Z	$9999.2 \pm 7.6 \text{ mm/min}$	$0.534 \pm 0.021 \text{ m/s}^2$	$-0.564 \pm 0.024 \text{ m/s}^2$
W	$4998.1 \pm 4.4 \text{ mm/min}$	$0.298 \pm 0.015 \text{ m/s}^2$	$-0.330 \pm 0.018 \text{ m/s}^2$
B	$719.6 \pm 3.6 \text{ }^\circ/\text{min}$	$0.063 \pm 0.003 \text{ }^\circ/\text{s}^2$	$-0.058 \pm 0.003 \text{ m/s}^2$

the machine Tab. 3.1. It is observed that accelerations and decelerations of the axes have the same absolute value. Since the EPC controls only the Cartesian axes X , Y , and Z , the acceleration of remaining drives W and B are neglected.

It is therefore assumed that the acceleration of the axes is 0.4 m/s^2 , thus being slightly conservative in the calculation of the optimal feed rate profiles.

6.1.2 Transients of feed rates and accelerations

When several feed values are sent to the NC in few seconds, the velocity profiles are dominated by transients. It is necessary to characterise how the NC handles these transients to design properly the velocity profiles calculated by the optimal control. Three might be the strategies followed by the NC to adapt online the velocity profiles when a new feed rate is set:

1. Constant acceleration: the axes are moved at the maximum acceleration. The time duration of the transient is variable and depends on the increment of speed applied.
2. Constant time: the acceleration of the axis is calculated from the increment of feed between two consecutive time instants.
3. Variable time and acceleration: the intensity and duration of the accelerations are variable and they are calculated online by the *look ahead* planner.

To identify the adopted strategy, X , Y , and Z are moved independently, by feed increments with amplitude increasing in time, Fig. 6.3. The test starts with a low feed, that is doubled every two seconds. On the X and Z axis the last speed increment is set to a value that is higher than the maximum feed supported, therefore the feed cannot follow the reference. From the acceleration plots (on the right) it is possible

to state that when the feed is modified, the NC applies accelerations for a constant period, but with different amplitudes. The time span required to accelerate the axes is 394.74 ± 57.29 ms. The uncertainty of the time durations is estimated by calculating the standard deviation among all the transients.

However, under step inputs the NC has to adapt the feed profiles only during a short time interval. To analyse how the system reacts to inputs continuously changing, the feed rate is changed with a ramp input.

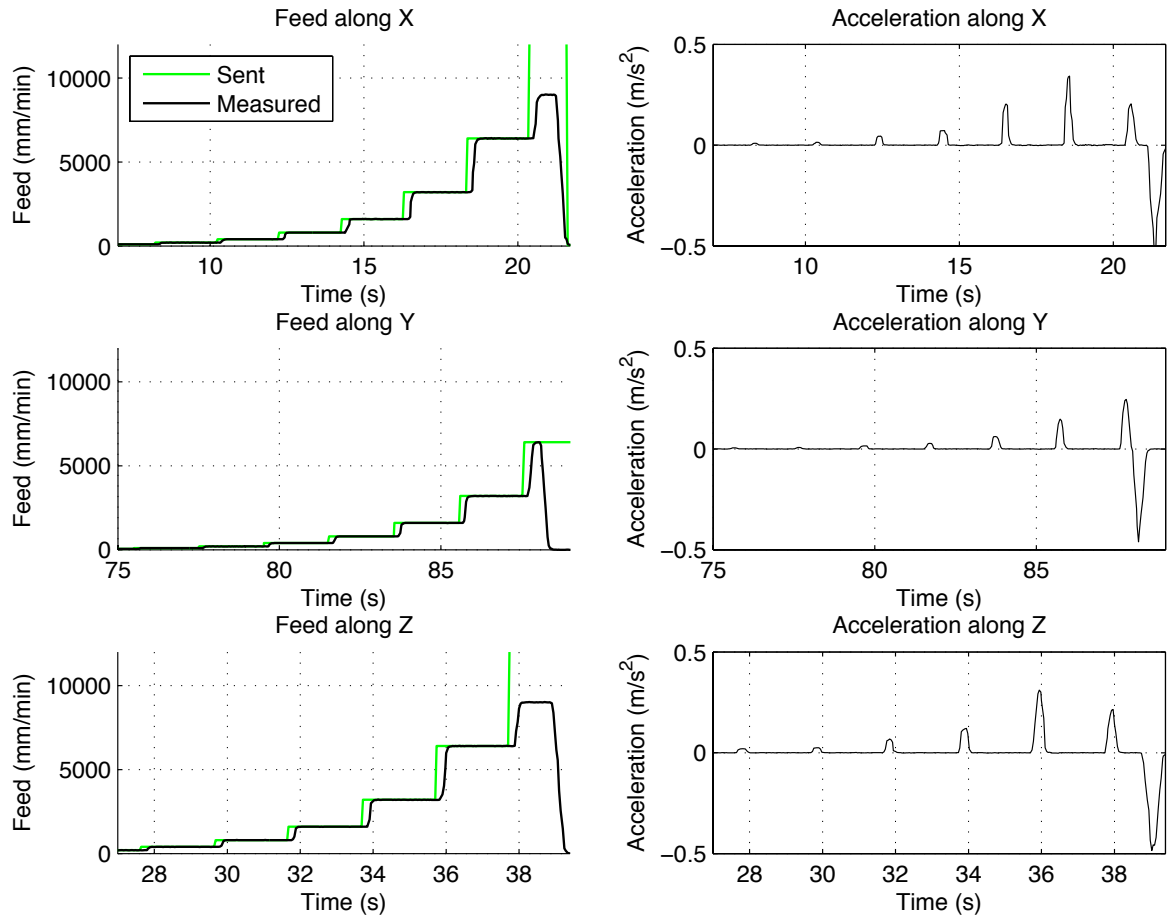


Figure 6.3: Tests performed to identify the duration of the acceleration transient.

As preliminary tests, this behaviour is studied by changing the feed rate with triangular sweep profiles. The height of the triangles is set equal to the maximum feed rate. The first period of the sweep is defined accordingly to the axis travel, and it is subsequently halved. The test along the Y axis is performed moving the cutting head from the top to the bottom and vice versa, to take into account the effect of inertia of the moved masses. It is observed that the deceleration of Y is higher when the cutting head is lowered, and the acceleration is higher when the cutting head is risen. This effect is also shown in Fig 6.4, where for the Y axis the measured feed profiles are slightly asymmetrical. Note that, due to the inertia of the axes, the phase displacement between the set and the measured feed becomes bigger as the period of the sweep is reduced. From this test is not possible to obtain useful information about the frequency response of the system because the traveled length is too short and the latency of the communication is too high. To avoid these problems another test is designed.

Only the deceleration of the axes is studied with this second experiment. The drives are slowed down starting from three different feed rates to zero, with different decelerations. The starting feed are 10000 mm/min , 6666 mm/min , and 3333 mm/min . The magnitude of the maximum deceleration is calculated through a linear regression of the the feed profile during the transients.

An example of the acquisition made along X for a deceleration of 0.325 m/s^2 is shown in Fig. 6.5. The red, green, and blue lines are the measured feeds for tests performed starting from 3333 mm/min , 6666 mm/min , and 10000 mm/min respectively. Note that the deceleration ramps are smoothed when the drives are decelerated. This is due to the fact that the NC generates velocity profiles that are jerk limited. The magenta line represents the set feed profile, calculated from the chosen deceleration. The black solid and dashed lines are the fit of the measured feeds, estimated taking into account only the data sampled during the feed transitions. The slope of the magenta line is therefore

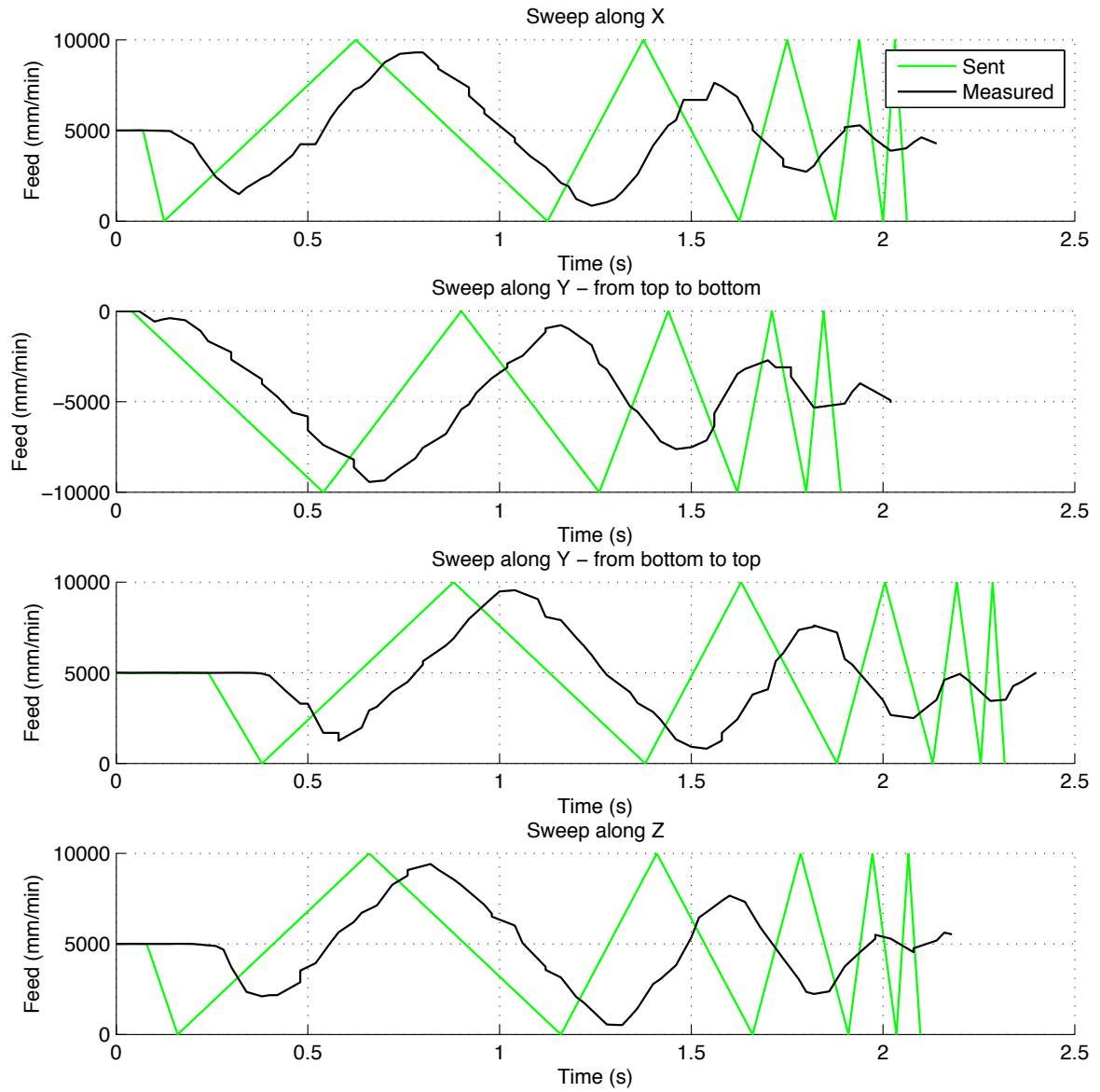


Figure 6.4: Set and measured sweep profiles on the feed rate. In the test performed along Y going from the top to the bottom the feed rate is negative because positive Y are upwards.

the measured deceleration. Qualitatively, as much similar are the slopes of the magenta and black lines, as better the drive can follow the feed reference, and so the imposed deceleration.

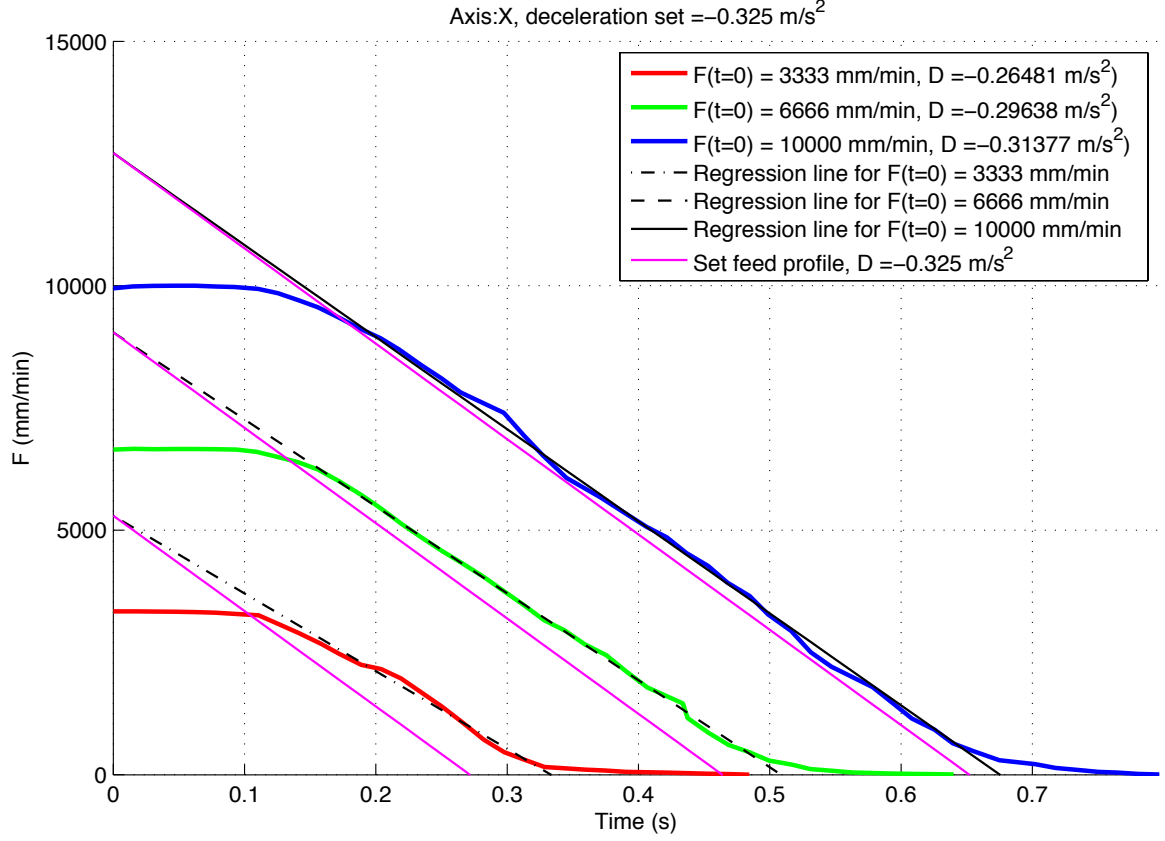


Figure 6.5: Example of a deceleration test performed starting from different initial feed rates.

In Fig. 6.6 are shown the data collected for all the performed tests. On the abscissa is reported the measured deceleration, while the set deceleration is on the ordinate. At a critical value the curves start to diverge, and the measured decelerations become smaller than the set ones, until they saturate to a maximum value. For the X and Y axes the value of the critical deceleration depend on the initial value of the feed, and the deflection points are shifted to lower accelerations, as lower is the initial feed rate. This is due to the fact that the NC does not apply instantaneously the feed rate passed by the EPC, but rather it interpolates the movements between the current feed and the desired one. For each test the percentage error in following the set deceleration

is calculated. The results obtained from the star input tests in Sec. 6.1.1 can be confirmed, scilicet that the maximum acceleration achievable under a feed ramp input is 0.4 m/s^2 , with a tracking error lower than 25%.

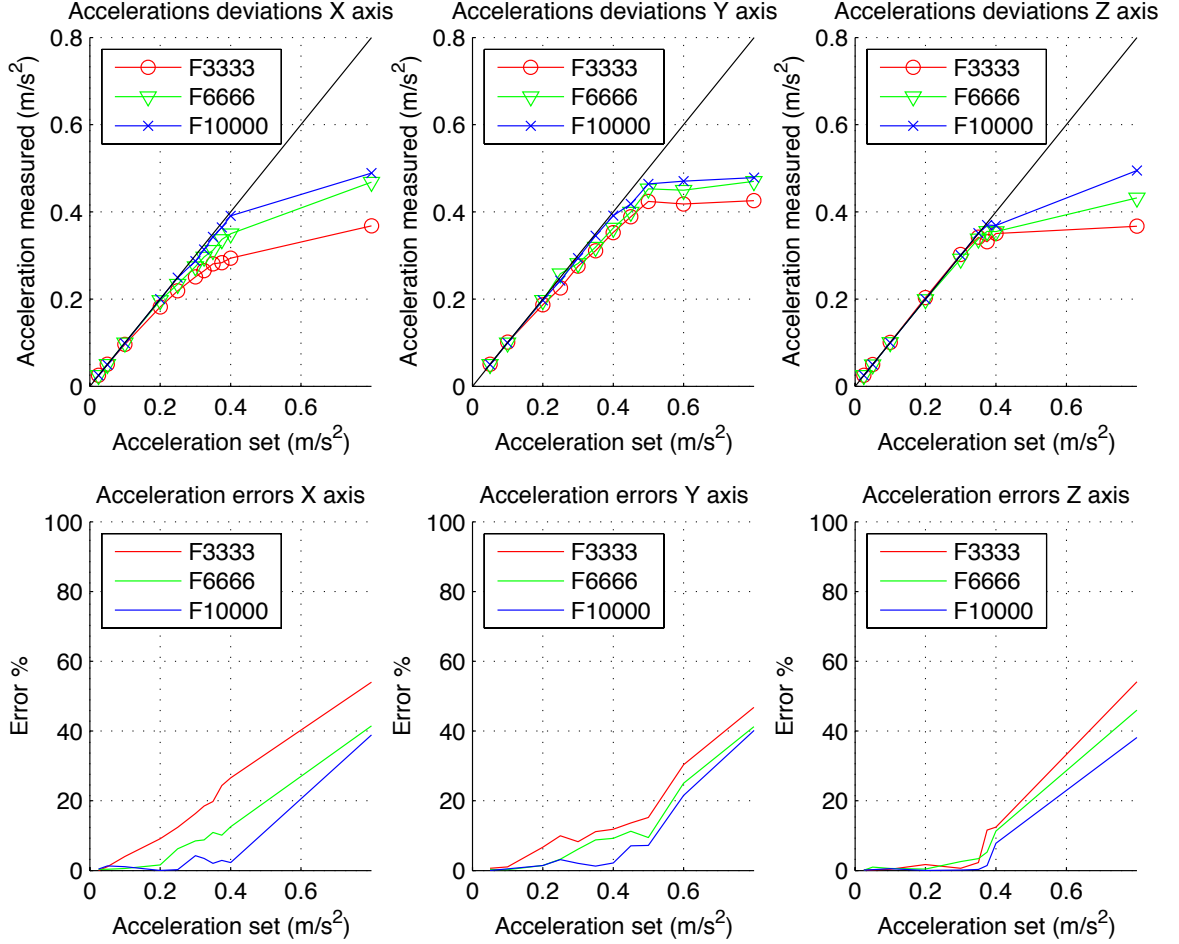


Figure 6.6: Results obtained from the deceleration tests for the X (left), Y (centre), and Z (right) axes.

6.2 Dynamics of the spindle

6.2.1 Spindle acceleration

The experimental tests here described are aimed at identifying the maximum rotational acceleration and deceleration of the spindle. The former has a lower module than the latter because the NC can decelerate the spindled through a mechanical brake. The model of the spindle integrated into the EPC assumes that the magnitude of the maximum rotational acceleration and deceleration of the spindle are equal. In order to ensure the feasibility of the optimal solution, the spindle acceleration and deceleration are set equal to the smallest between the two.

A step pyramid on the spindle speed is imposed and the mechanical torque is measured. The pyramid goes from 100 rpm to 1500 rpm, with increments of 100 rpm. The tests are repeated three times to perform statistical analysis on the measurements. Each step lasts 20 s to ensure the attainment of the steady state conditions.

In Fig. 6.7 are reported the measurements collected during one test. The spindle speed exhibits jerks because it is measured from the feedback loop controller on the spindle speed. It should be mentioned that the tool tip rotates with a more regular speed, due to the inertia and frictions of the kinematic chain bringing the movement from the electric motor to the ram. The spindle acceleration is calculated with a finite difference scheme after the data have been filtered with moving mean whose period is of 5 samples. The average maximum acceleration is 123.5 rad/s^2 . Shapiro's test on the acceleration data give a p-value of 0.47, therefore a Student's t-test is used to estimate the confidence interval $[110.8, 136.2] \text{ rad/s}^2$, with a confidence level of 95%. In this test the null hypothesis is: measured mean equal to zero.

When modelling the spindle, it is assumed a maximum acceleration of 104.7 rad/s^2

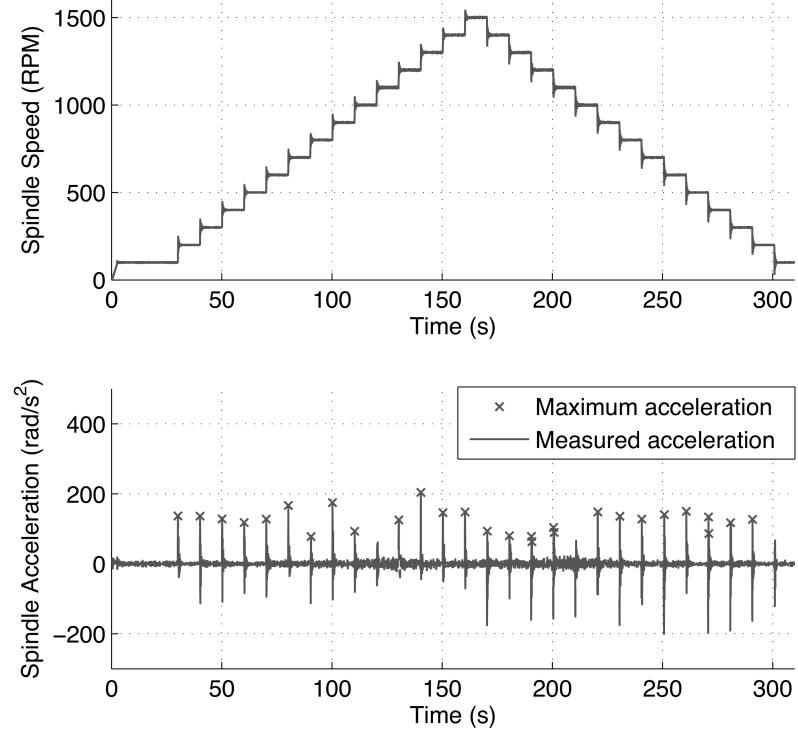


Figure 6.7: Measured spindle speed (top), and calculated accelerations (bottom) for one of the three tests performed.

($\approx 1000 \text{ rpm/s}$), thus being slightly conservative and taking into account the high uncertainty of measurement.

6.2.2 Mechanical torque

The mechanical frictions in the kinematic chain of the cutting head are estimated by measuring the free run mechanical torque. These frictions are due to the contact between gears, bearings, and viscosity of the lubricant oil. The mean mechanical torque is measured at different steady state spindle speeds (Fig. 6.7), and it is repeated for the ram (W axis) at different positions. The ram is fully retracted when $W = 0 \text{ mm}$, and it is fully extended when $W = 300 \text{ mm}$. As can be seen in Fig. 6.8, the free run

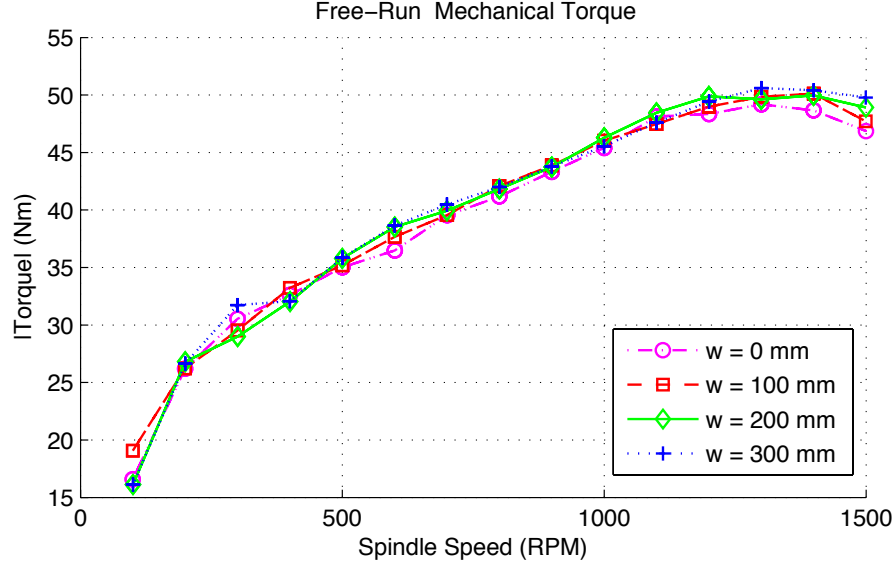


Figure 6.8: Measured free run mechanical torque at different positions of the ram.

torque is not affected by the position of the ram. Moreover, the free run torque is non linearly dependent from the spindle speed. A robust model of the friction torque with respect to residuals is obtained by interpolating with a second order polynomial the free run torque in function of the spindle speed. In the adopted model, the constant term can be associated to the Coulomb friction T_c , the linear term to the viscous friction T_v , but the quadratic term T_2 has not physical meaning. The latter might be wrongly associated to the aerodynamic drag by virtue of its dependency from the quadratic velocity. However, aerodynamic forces are generated only when a relative wind is impacting against a body, that is not the case for the milling process. Moreover, the drag torque should quadratically increase with the spindle speed. The quadratic term in the regression is therefore included because it allows to obtain residuals that are independent from the fitted spindle speed, Fig. 6.9. The expression of the free run

torque T_{fr} is:

$$T_{fr} = T_2 + T_v + T_c = -1.629 \cdot 10^{-5} S^2 + 0.047 S + 15.670 \quad (6.1)$$

where S is the spindle speed in RPM. The corrected mechanical torque T_{cm} is obtained subtracting T_{fr} to the raw measurements:

$$T_{cm} = T_{raw} - T_{fr} \quad (6.2)$$

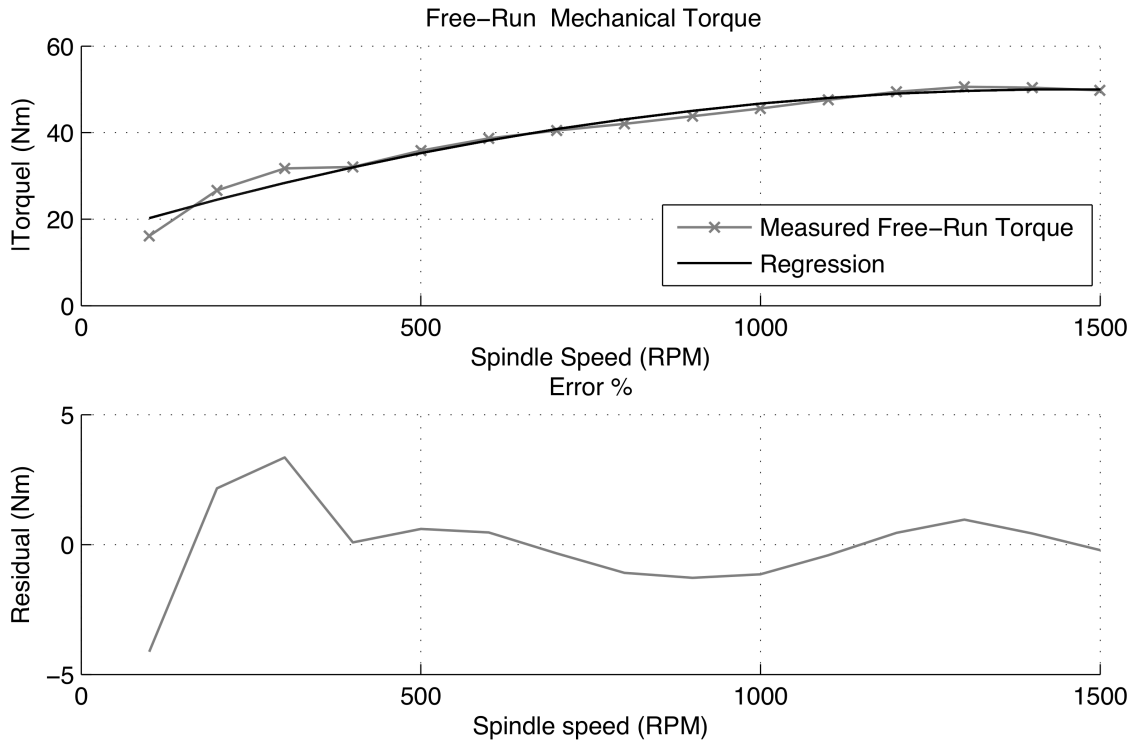


Figure 6.9: Measured and fitted free run torque for the test with the ram fully extended (top), and residuals of the regression (bottom).

Validation

From some preliminary tests it is observed that —due to a bad scaling of the torque measured from the CN— the measured cutting power is higher than the absorbed power current. This issue is addressed by identifying the correct scaling factor, through tests in which the spindle is continuously accelerated and decelerated with four different accelerations. By knowing the acceleration and the moment of inertia of all the shafts in the kinematic chain of the cutting head, it is possible to estimate the mechanical torque generated by the motor.

Given an acceleration at the spindle, the dynamics of the remaining shafts are calculated by analysing the kinematic chain through the *depth-first search algorithm* (103). The theoretical mechanical torque at the motor T_{em} is calculated by summing the inertial torque of each shaft in the kinematic chain:

$$T_{em} = \sum_{i=1}^{N_s} \dot{\omega}_i J_i \quad (6.3)$$

where N_s is the total number of shafts, while J_k and $\dot{\omega}_k$ are the moment of inertia and rotational acceleration of the i -th shaft (see Tab. 3.3). The scaling factor is given by the ratio between T_{em} and the badly-scaled measured torque T_{cm} :

$$K = \frac{T_{em}}{T_{cm}} \quad (6.4)$$

The average scaling factor calculated among all the tests is $K = 0.67$, Fig. 6.10. The corrected measured mechanical torque T_m is obtained by subtracting the free run torque to T_{raw} , and by multiplying the result for the correction factor K and for the mechanical efficiency of the transmission system $\eta_m = 0.85$ given by the machine manufacturer:

$$T_m = (T_{raw} - T_{fr})K \eta_m \quad (6.5)$$

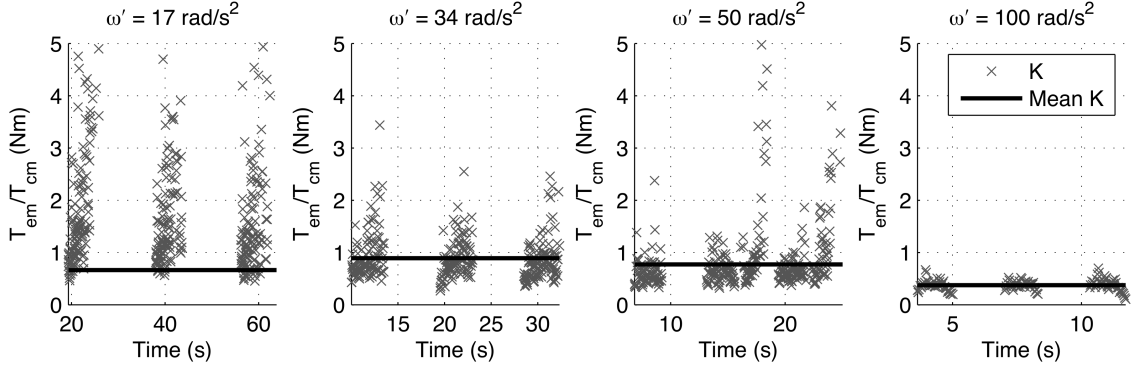


Figure 6.10: Scaling factor K estimated for four acceleration/deceleration tests.

The torque measurements are validated by cutting a workpiece with a known specific energy. A face milling is performed with a cutting depth of 1.3 mm. The mill has 5 flutes and a diameter of 63 mm. It is cut the *AA6082-T6* with a specific energy of 0.75 J/mm^3 . As can be observed in Fig. 6.11, the measured and the theoretical values of the mechanical torque are in agreement. Anomalous torque peaks at the beginning and end of the cut are due to the fact that the measured signal comes from the feedback loop on the spindle speed. When the mill enters in contact with the workpiece, the spindle tends to be slowed down, therefore the feedback controller compensates this deviation from the reference speed by increasing the current, and so the torque, at the spindle.

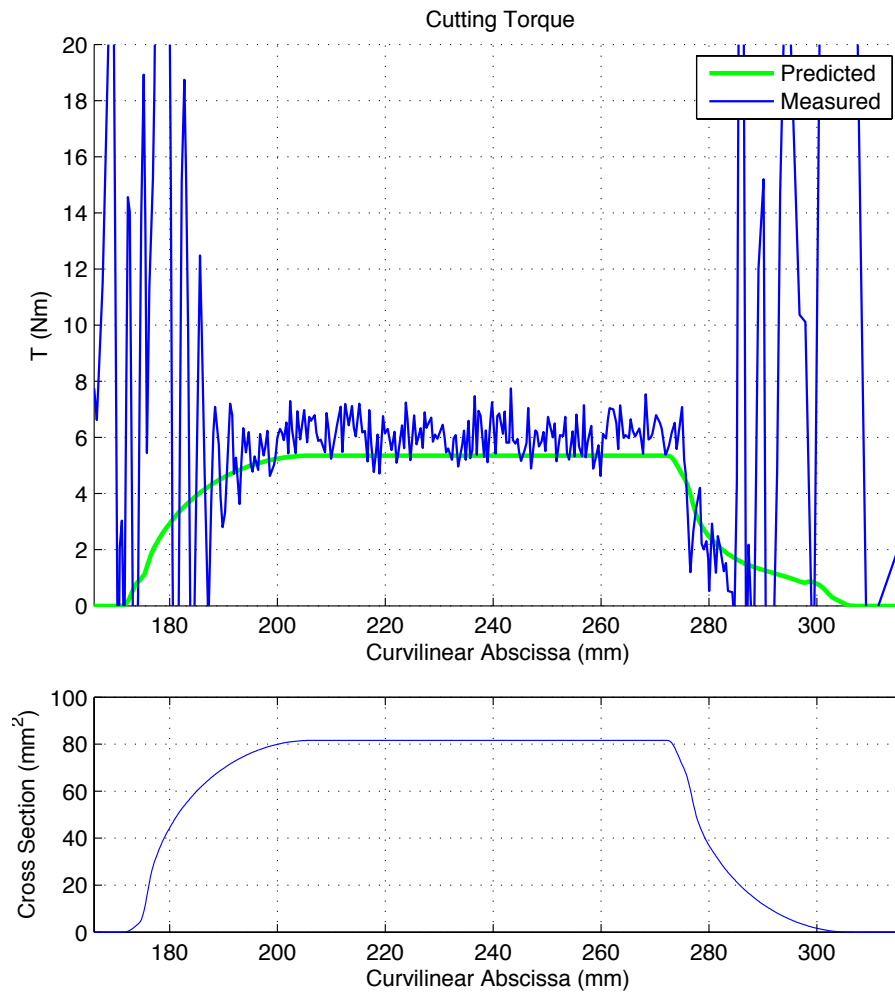


Figure 6.11: Measured mechanical torque (top), and cut cross section (bottom).

6.3 Dynamic compliance

In order to analyse the stability of the process, the dynamical response of the system (machine, tool, and workpiece) is described through the *Frequency Response Function*. The FRF should cover a frequency range enough wide to include all the main modes of the structure. The shape of the stability lobes diagram is indeed given not only by the stability pockets generated by each mode, but also by the intersection of lobes associated to different modes. If a mode is missing in the FRFs, then the lobes diagram will appear distorted or will not describe the true stability space at all.

The system is modelled as Multiple Input Multiple Output (MIMO). The inputs are the forces on the tool tip, while outputs are the displacements, therefore the FRF of the system is represented by a 3×3 matrix:

$$FRF = \begin{bmatrix} FRF_{x,x} & FRF_{y,x} & FRF_{z,x} \\ FRF_{x,y} & FRF_{y,y} & FRF_{z,y} \\ FRF_{x,z} & FRF_{y,z} & FRF_{z,z} \end{bmatrix} \quad (6.6)$$

The notation is the following: the first subscript indicates the direction along which the force is applied, while the second subscript is used to specify the direction along which the displacement is measured.

In this study impact tests are performed to measure the FRF matrix. The ram of the machine is excited through a *PCB 086D20* hammer with a *084A63* hard plastic tip. A three-axial *PCB T356A15* accelerometer is mounted on the tool tip and it is acquired at 5000 [Hz], Fig. 6.12. The impacts are given in a position close to the accelerometer, and are repeated seven times in order to reduce the uncertainty of measurement associated to its variable intensity and orientation. The same procedure is applied to identify the FRF of the workpiece and working table, Fig. 6.13.

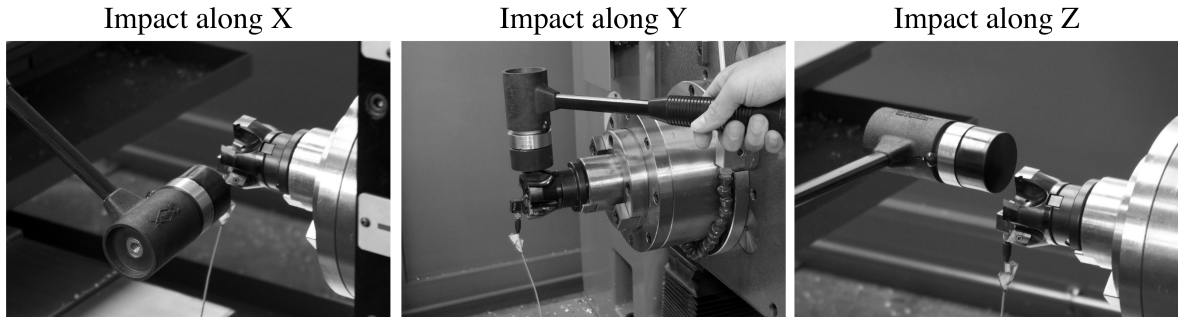


Figure 6.12: Direction of impacts on the tool.



Figure 6.13: Direction of impacts on the workpiece.

The software used for acquisition is *CUTPRO MalTF*, which calculates the *Frequency Response Function* (FRF) of the machine with the procedure proposed by Altintas (96). Sampled force and acceleration signals are multiplied by an exponentially decaying window, thus neglecting the data associated to a null force. The exponential window decays from unity to 0.05 in the total sample time of 10 s. The Fourier transform of the displacement is calculated by integrating twice the Fourier transform of the accelerations. Finally, the FRFs are computed through the cross-power spectrum of measured signals, and the obtained response is validated by verifying that the coherence function is unitary.

The homogeneity of dynamic properties of the machine and the workpiece across the

Table 6.2: Positions at which the FRF of the tool are tested, expressed in the absolute frame of reference of the machine.

Test	X mm	Y mm	Z mm	W mm	B °
A_t	-350.3	-250.3	+350.3	+0.3	0
B_t	-350.3	+250.3	+350.3	+0.3	0
C_t	-350.3	+250.3	+350.3	+300.3	0
D_t	-350.3	-250.3	+350.3	+300.3	0

working volume are verified by changing and combining the positions of linear axes accordingly to a design matrix (104). The real part of the recorded FRFs for different position of the axes are in Fig. 6.14. As expected, the direct FRFs (on the diagonal of the figures) are higher in magnitude with respect to the transversal ones. The workpiece can be considered stiff since all its FRFs are close to zero.

The compliance of the whole system is independent from the positions of the X , Y , and Z axes. Only the position of the ram, that is the W axis, affects the frequency response of the ram. In Fig. 6.15 are reported, for four positions of the ram, the real and imaginary part of the measured direct FRFs. When the ram is retracted ($W = 0$ mm) the first mode is around 850 Hz and low. In this case the system can be considered stiff. On contrary, when the ram is fully extended, ($W = -300$ mm) the first mode is dominant and at a lower frequency, approximately 200 Hz.

6.3.1 System identification

In order to compute lobes diagrams that are accurate across the whole working volume, it must be characterised the influence of the position of the ram on the dynamical properties of the machine. The FRF of the ram are measured at different position of the W axis, which is changed from -220 mm to -300 mm with increments of 5 mm. A robust

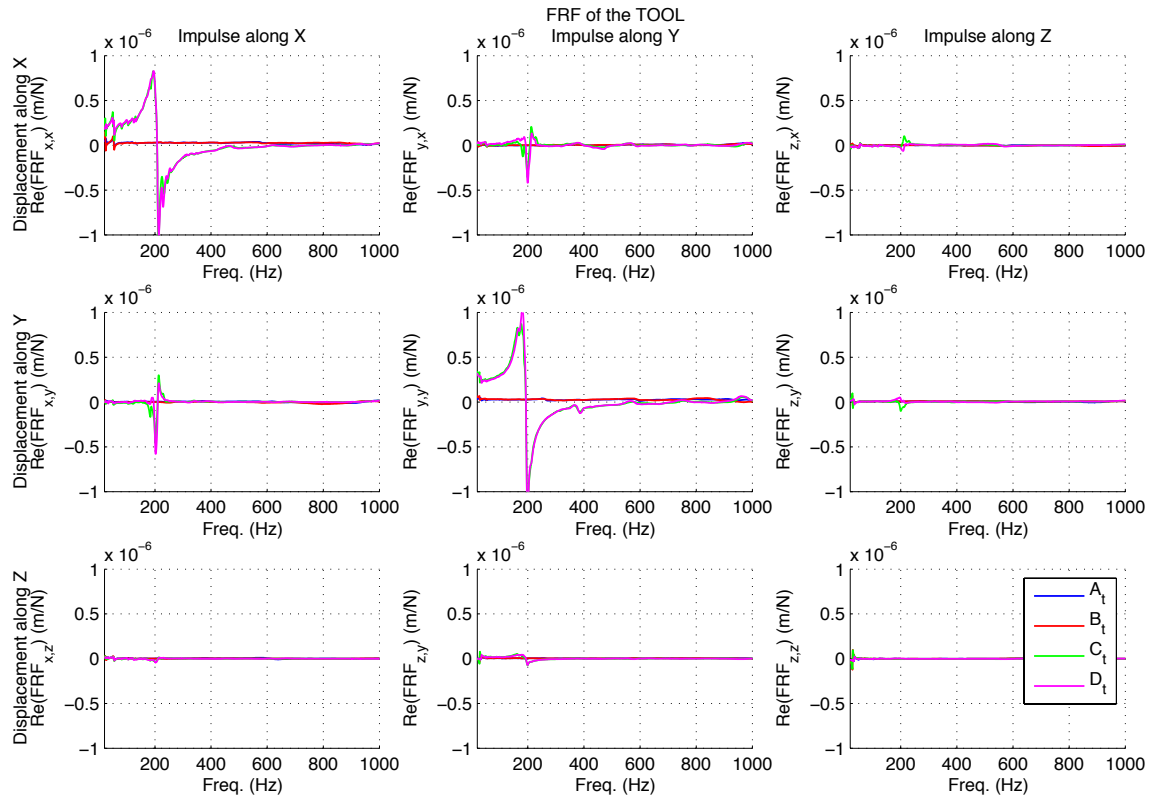


Figure 6.14: Real part of the FRF matrix for the impacts on the tool, at different positions of the axes (see Tab. 6.2).

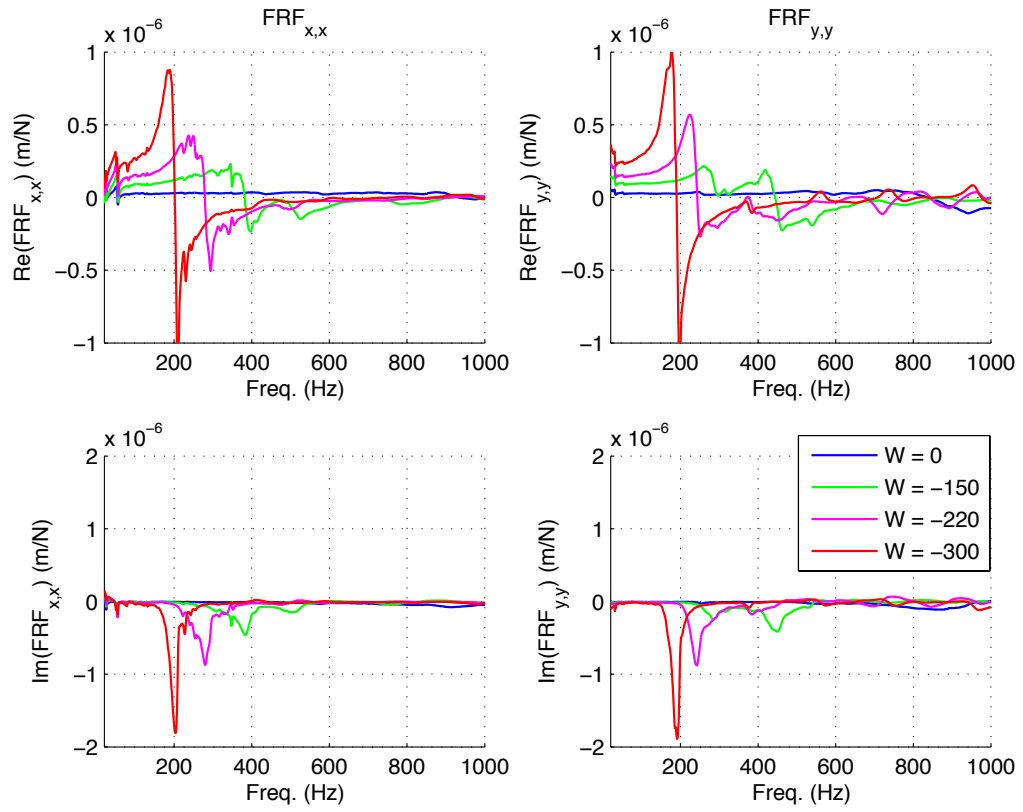


Figure 6.15: Real and imaginary part of direct FRF along X (left) and Y (right), at different positions of the ram.

calculation of the lobes diagram at intermediate positions is performed by generating a model of the dynamic compliance of the ram in function of W . This operation can be performed with two approaches: system identification and regression.

When identifying a dynamic system, a model is defined a priori and its coefficients are estimated by fitting the model to real data. The identified coefficients describe a physical property of the system (mass, stiffness, damping...). Different approaches are available in literature for system identification, and a comprehensive review was given by Kerschen *et al.* (105). In this study several well-known strategies have been tested, such as: lumped model inverse analysis (106), Linear Parameter-Varying (LPV) modelling (107), pole/residue parametrisation (108) and estimation of state-space model using a subspace method (109). Nevertheless, none of them is enough accurate to catch all the modes in the FRFs that concur to form the lobes diagram.

For this reason, a regression is carried out on the acquired data. The difference between regression and system identification is that in the former the coefficients of the model don't have necessarily a physical meaning. Autoregressive exogenous (ARX) model (110) and estimation of Output-Error polynomial model (111) have been fit to the data. However, in both cases it has been observed that a low order polynomial is not enough accurate, while a high order polynomial is affected by over-fitting.

The problem of regression and classification is well known in the field of machine learning, where large amount of data must be analysed. Classification is usually performed by means of *K-mean* algorithm and *Support Vector Machines* (SVM) (112). Regression can be still performed by using SVM, but more efficient algorithms are Artificial Neural Networks (ANN) (103), Locally Weighted Projection Regression (LWPR) (113), and Gaussian Processes (GP) (114).

ANN were used in the past 60 years in a wide range of applications, however the correct

design of their architecture is still a complex task. The main drawback of these models is their lack of explanation capability (115). ANNs are considered "black-box" since it is difficult to interpret their output and the effect of the parameters of the network on its performance.

LWPR can be used to fit non linear function in high dimensional spaces with redundant and irrelevant input dimensions. This method is particularly suitable for large amount of data, which can be learnt online through an incremental learning. In this study the dynamic behaviour of the machine is described by few data points, which are collected beforehand the process optimisation. For such conditions LWPR loses its edge against the generalisation capability of GP.

6.3.2 Regression of the dynamic compliance

Here are summarised the results obtained with the regression of the FRFs through GP. The code is implemented in Matlab by exploiting the open source GPML Toolbox v3.2 (116). The 2D dynamical model of the machine used to compute the lobes diagram takes into account the behaviour of the system along X and Y axes. Thanks to this simplification, only the elements of the FRF matrix associated to these directions are regressed, and those associated to Z are neglected. Moreover, from preliminary tests it has been observed that the computation of the lobes diagram is affected mainly by the direct FRFs, meaning that only two of its components have to be regressed: $FRF_{x,x}$ and $FRF_{y,y}$. These two elements are represented as vectors of complex numbers, which are split into real and imaginary parts. Therefore, in total four GPs are trained.

The formulation of the GP is the one described in the Appendix. A. As remarked in the previous chapter, the regression given by GP is closely related to the behaviour of the data rather than to a parametric function. It means that different functions can be described by the same hyperparameters as long as the different sets of training data

exhibit similar probabilistic characteristics (i.e. a null mean, and a similar covariance). In Tab. 6.3 are reported results of the trainings of the four FRFS. Note that it is possible to generalise the set of hyperparameters, thus performing the training only for one set of data. In this study are used the hyperparameters found by training the GP with the real part of $FRF_{x,x}$.

An example of the regression is reported in Fig. 6.16 for the real part of $FRF_{y,y}$. On the left it is shown the measured FRF (top) and its contour plot (bottom), on the right it is shown the regressed FRF. The FRF plot of the measurement appears to be quantised, while the regressed FRF is more continuous since it has been evaluated for a more dense grid of ram positions and frequencies.

The prediction is in agreement with the measured data. It can be observed that a main mode dominates the dynamics of the machine. As the ram is extended (W is decreased to -300 mm) the main mode moves to lower frequencies and its amplitude increases. This means that the machine becomes less stiff and vibrates more. A second lower mode is slightly visible at 380 Hz.

6.3.3 Validation of the regression

The accuracy of the regression achieved through GP has a direct impact on the reliability of the calculated stability lobes diagrams, see Appendix B. The stability lobes

Table 6.3: Trained hyperparameters. $\|nlml\|$ is the module of the negative log marginal likelihood, the higher it is, the more accurate the prediction is.

Hyperparameters	l	σ_f	σ_d	σ_n	$\ nlml\ $
$\text{Re}(FRF_{x,x})$	9.0	$1.6 \cdot 10^{-7} \text{ m/N}$	$2.2 \cdot 10^{-8} \text{ m/N}$	$4.2 \cdot 10^{-9} \text{ mN/}$	23720
$\text{Im}(FRF_{x,x})$	8.8	$1.6 \cdot 10^{-7} \text{ m/N}$	$2.0 \cdot 10^{-8} \text{ m/N}$	$4.1 \cdot 10^{-9} \text{ mN/}$	23777
$\text{Re}(FRF_{y,y})$	10.2	$1.7 \cdot 10^{-7} \text{ m/N}$	$2.1 \cdot 10^{-8} \text{ m/N}$	$4.1 \cdot 10^{-9} \text{ mN/}$	23866
$\text{Im}(FRF_{y,y})$	10.3	$1.7 \cdot 10^{-7} \text{ m/N}$	$2.2 \cdot 10^{-8} \text{ m/N}$	$4.1 \cdot 10^{-9} \text{ mN/}$	23861

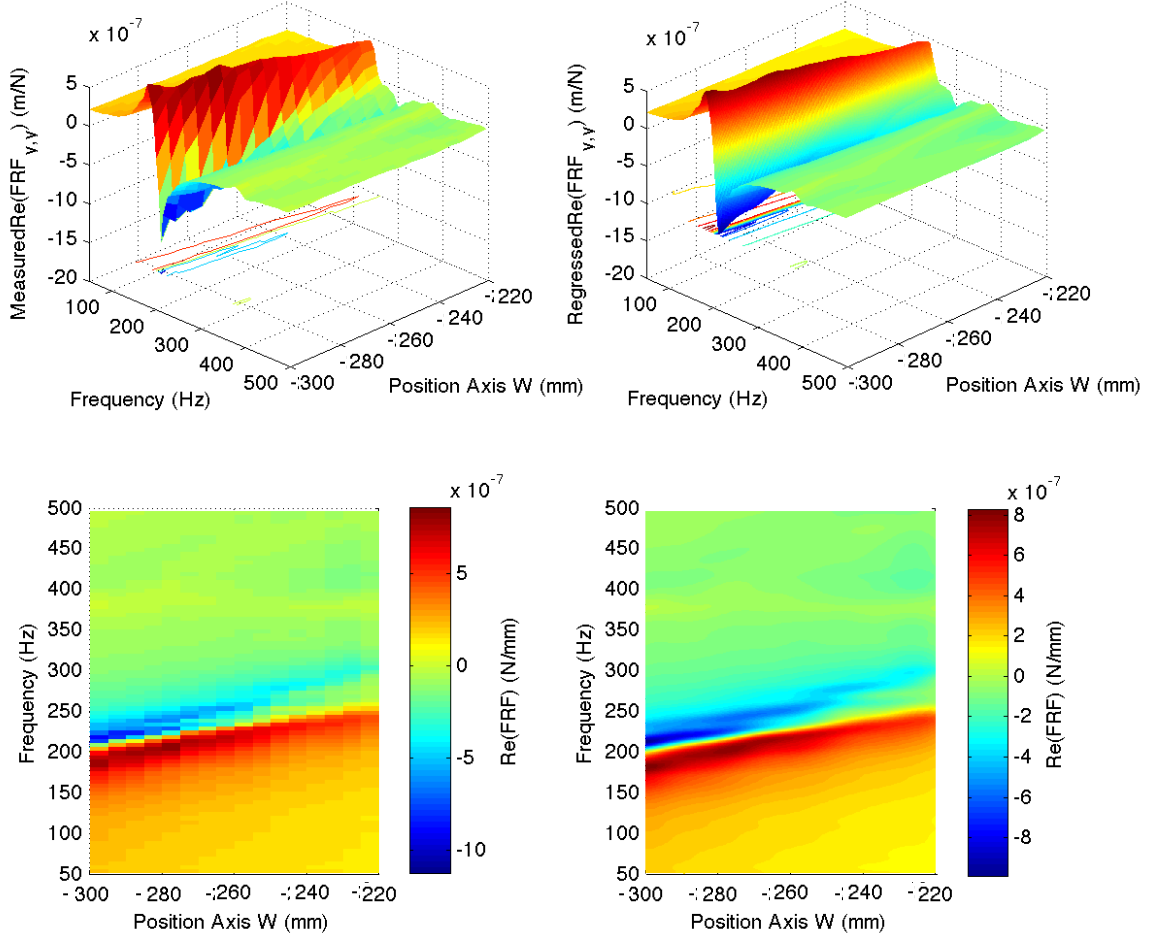


Figure 6.16: Real part of the measured (left) and regressed (right) $FRF_{y,y}$.

diagrams are calculated by considering as reference process the full immersion face milling, since for this cut the calculation of the engagement arcs is simplified: the radial immersion is between 0° and 180° . The mill has a diameter of 63 mm, with 5 WIDIA XPHT160408-AL inserts. The cut material is the AA 6082-T6 with cutting constants of $K_{rc} = 264.9 \text{ N/mm}^2$ and $K_{tc} = 614.1 \text{ N/mm}^2$. These values have been identified as explained in Appendix D. The process damping is set to zero.

In Fig. 6.18 are compared the stability lobes diagram calculated along the W axis with the measured and with the regressed FRFs, and it is observed that the two stability

lobes surfaces are in agreement. When W is set to bigger values (i.e. the ram was shorter), the stability limit is at a higher cutting depth and the stability pockets move to higher spindle speeds, meaning that the machine becomes more stiff.

To better interpret the stability lobes surfaces they are sliced by fixing W to a specific value, e.g. to -220 mm in Fig. 6.17. This is the condition at which the machine is more stiff and exhibits many different strong eigenmodes. In such complicate situation the stability lobes diagram tends to be complicated, but it is still well predicted by the FRFs regressed through GP.

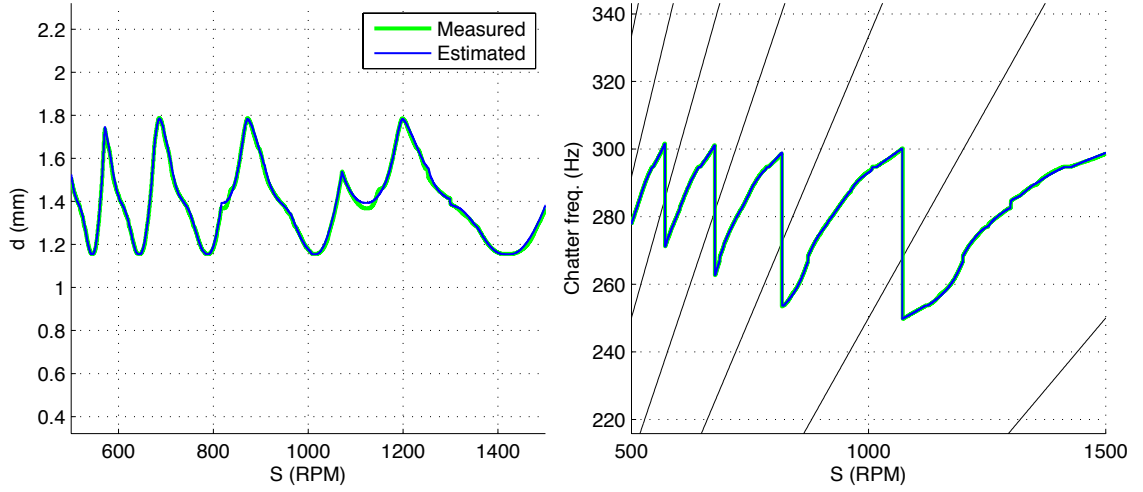


Figure 6.17: Stability lobes diagram calculated for the ram at the position -220 mm (left) and chatter frequencies (right).

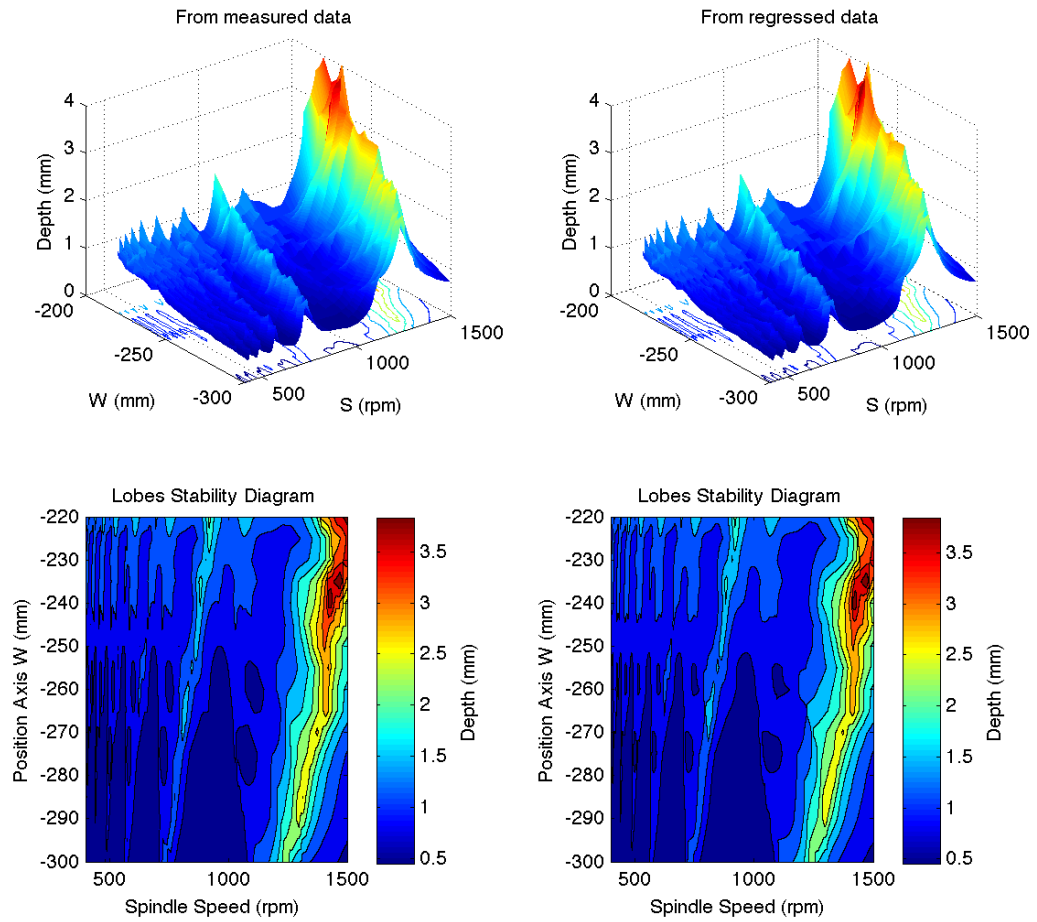


Figure 6.18: Stability lobes diagrams calculated from measured FRFs (left), and from regressed FRFs (right).

6.3.4 Validation of the stability lobes diagrams

The stability lobes diagrams are validated by recording the vibrations of the cutting head during cuts carried out at different spindle speeds and cutting depths. Two positions of the ram are considered during the tests: $W = -300$ mm and $W = -270$ mm. The triaxial accelerometer Dytran 3213M6 is acquired at 10 kHz, and it is screwed directly on the cutting head, as much close as possible to the tool, Fig. 6.19. The transmittance between the tool tip and the position of the accelerometer is neglected, since the aim of these tests is only to identify the chatter frequency (and not the amplitude of vibrations). The choice of spindle speed and depth of cut to be tested is performed according to the automatic methodology proposed by Bediaga *et al.* (117). In this approach the boundary of the SLD is searched by iteratively adjusting the cutting depth and the spindle speed. The first cuts are aimed at identifying the location of an instability pocket in the SLD. The cutting depth is low in the beginning, and it is increased until chatter is detected. The spindle speed is then adjusted in a such way that the new tooth passing frequency is equal to a multiple value of the chatter frequency. Once a stable cut is achieved, the cutting depth is increased again and the iterative process is repeated. The tests are interrupted when the chatter persists after five consecutive iterations, since that condition means that the maximum stable depth has been overshoot.

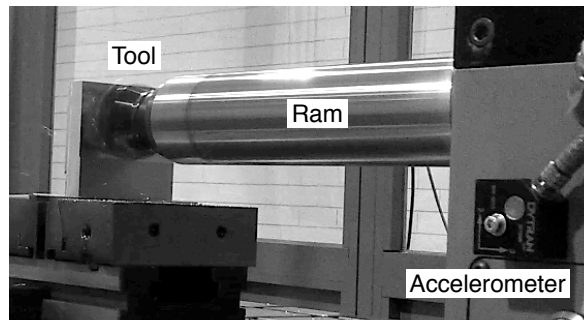


Figure 6.19: Position of the accelerometer.

The chatter onset is quantified by calculating the *energy ratio chatter indicator* (CI_{ER}) (118):

$$CI_{ER} = \frac{E_c}{E} = 1 - \frac{E_p + E_n}{E} \quad (6.7)$$

where CI_{ER} is given by integral quantities related to the Power Spectrum Density (PSD) (119) of: recorded accelerations (E), asynchronous component of the signal related to the chatter (E_c), synchronous components associated to the forced vibrations (E_p), and noise (E_n). All the PSD are calculated on signals that are windowed with a Hamming window (120).

High CI_{ER} are given by unstable cuts. Chatter indeed is associated to phase shift of half-period between the vibration of the tool and the tool pass frequency. When this phase shift occurs, the integral of the PSD of E_p becomes smaller, while the integral of the PSD of E_c increases.

In Fig. 6.20 are presented the SLD for the two positions of the ram tested. Each point in the SLD has a colour and a dimension that is related to the measured CI_{ER} ; the bigger and more red the point is, the more severe the chatter is. It can be observed that when $W = -300$ mm, the amplitude of the predicted stability pockets is in agreement with the measurements, but they are slightly shifted to higher spindle speed. This is due to the fact that the SLD have been measured and validated in different moments of the day, during which the temperature and lubrication of the machine has changed. For $W = -270$ mm the abscissa positions of the stability pockets are well predicted, but their amplitude is over estimated by the calculated SLD. With the ram at this position the machine is sensibly more stiff, therefore the FRF are affected by a lower signal-to-noise ratio. Nonetheless, it should be remarked that in the strategy adopted for integrating the information of SLD into the OCP it is more important to estimate accurately the abscissa position of the stability pockets rather than their amplitude (see Chapter 2.4.2). It is therefore concluded that the calculated SLD are sufficiently in agreement with the measurements made.

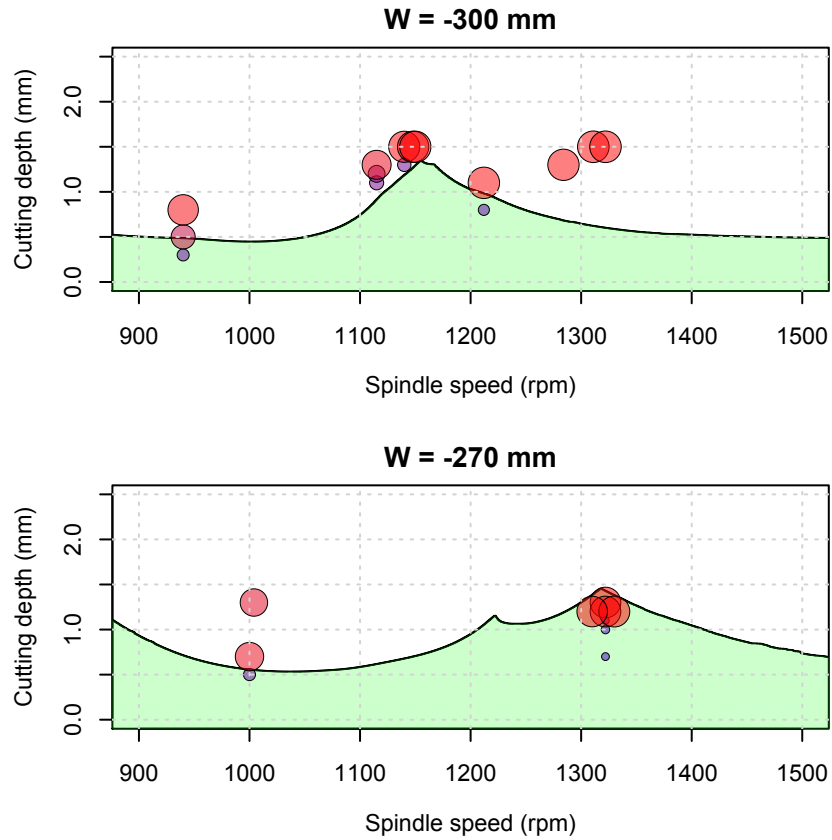


Figure 6.20: Stability lobes diagrams validated at two positions of the ram: -300 mm (top) and -270 mm (bottom). The size of the circles is proportional to the calculated CI_{ER} .

Chapter 7

Applications of Augmented Reality for Safe Manufacturing

In this chapter it is proposed an approach to enhance the sensing layer of the EPC through the integration of images and videos recorded by cameras. The aim of this study is to state the feasibility of 3D scene reconstruction of the working volume through 3D depth cameras. Several issues must be addressed when applying image processing techniques to augment the reality of milling processes, such as: the presence of high frequency vibrations, debris, chips, coolant liquids inside the working volume, as well as reflecting surfaces of the tool and workpiece. Image processing system can be exploited in numerous applications thanks to their flexibility. The information displayed to the CNC user can be augmented by integrating data from the NC, process simulators, and auxiliary safety systems.

7.1 State of the art

In the past years several authors applied the AR in design and manufacturing (121; 7). Nevertheless, up to now the efforts made to augment the information collected from the machining processes are limited to few studies, whose main purpose has been to design efficient tool paths, and to ease the training of students and novice workmen. Weinert et al. (122) used AR to test off-line the tool path by including a virtual tool and a virtual workpiece in the video acquisitions of the real working volume. The system exploited a monocular camera and tracked the position of the tool holder by localising a marker sticked on top of it. Zhang et al. (123; 124) predicted offline the cutting forces and collisions by cutting a virtual workpiece with virtual tool. Online, the virtual model was aligned to the real working table. The tracking of the real tool was made with a monocular camera, by means of template matching. The system was used also to superimpose to the video acquisition of the machining volume the information about actual cutting forces, and active block. Olwal et al. (125) developed a system called ASTOR, an auto-stereoscopic optical see-through spatial AR system, to project information about the cutting forces, process parameters and G-code on the safety glass of the machine.

The main limitation of the approaches here reviewed is the same of collision avoidance performed by CAM softwares: they rely on a virtual model of the bulk workpiece that has to be generated with sufficient accuracy. In this study it is carried out a feasibility analysis to try to overcome this drawback through state of the art algorithms and hardware in 3D reconstruction. The geometry of the raw workpiece and tool can be learnt offline, directly from the scene, thus eliminating the error-prone human factor from the tool path safeness verification. It is moreover proposed the conceptual design of a collision avoidance system for safe tool paths based 3D dense scene reconstruction.

7.2 3D scene reconstruction

Complex tool paths and workpieces force CNC users to carefully design and check on air the programmed G-codes. However, the tool path verification is a time consuming task that often, for the sake of productivity, is neglected. The problem of online collision monitoring and avoidance is well known in many robotic applications where mobile robots have to travel across an unstructured environment, without to impact with objects that can be static (walls, columns, etc..) or moving (people, animals, other robots). In machining process the environment is partially structured since—in the worst scenario—only the raw workpiece has an unknown shape. However, more attention has to be given to the definition of collision in machining processes. The tool has to touch the workpiece to cut it, hence the contact between them is unavoidable, nevertheless three undesired conditions can be identified:

- The tool hits the workpiece without spinning or during rapid movements.
- The tool hits the working table.
- The tool holder hits the workpiece or working table.

The system developed to supervise the working area should be able to classify dangerous situations on the basis of an *extended set of features*, reconstructed from the status of the process, and relative position of tool and workpiece. The former is already gathered by the simulator module of the EPC, while the latter might be obtained by aligning a *virtual reality* (VR) model of the simulated process with a 3D set of points acquired from the real scene. To extend the knowledge of the real world by merging

measured and virtual data is the main paradigm of the *Augmented Reality* (AR).

In order to obtain a reliable 3D model of the scene, different camera poses must be acquired and fused together. The *Simultaneous Localisation And Mapping* (SLAM) technique can be used to obtain a map of the environment and the poses of the camera with respect to the reconstructed scene. At the core of the SLAM there is the *registration* process that allows to combine different spatial data sets in a single global model, thus generating large, complex, and coherent virtual scenes, while mitigating the effects associated to the measurement uncertainty.

In a virtual environment, objects can be represented by a set of points, namely *point clouds*. The most diffused algorithms developed for the registration of 2D and 3D point clouds are the *Iterated Closest Point* (ICP), and the *Normal Distribution Transform* (NDT) (126).

The ICP extracts key points from the images and then aligns two point clouds by minimising a metric that can be defined as: the point-to-point Euclidean distance (127), the point-to-plane distance (128), or the plane-to-plane distance of Mitra *et al.* (129). The ICP is a fast method, suitable for real-time applications, and it has been used in many projects for real-time 3D dense reconstruction (130; 131; 132). One of the first robust SLAM algorithms was developed by Davison (133), who registered sets of key points extracted from images through the ICP. This approach resulted to be robust with respect to scaling, rotation, translation and affine transforms. Klein *et al.* (134) paralleled the feature tracking and mapping processes by distributing the SLAM computation on a dual core computer, thus increasing the real time capabilities of the system. As a drawback, the ICP requires an explicit correspondence between the entities in the two clouds, otherwise the convergence of the registration would be slow, or not reached at all. This implies that slow movements should occur between two consecutive acquisitions, and the images should be highly textured.

When the spatial distribution of the two point clouds are not homogeneous and their density are not comparable, the assumption of explicit correspondence cannot be satisfied and the ICP would fail. In such cases it is possible to utilise the NDT (Appendix C). This algorithm exploits the normal distribution to generate a probabilistic model of the acquired scene, and it finds the transformation between two acquisitions by maximising the probability that the new point cloud is part of the statistical model of the environment. This method does not require to establish a direct correspondence between primitives (points, lines, or surfaces) therefore it is more robust to affine transformations than the ICP. As instance, the NDT has been applied successfully to align the CAD model and 3D measured shape of a sheet deformed through the Single Point Incremental Forming, in order to evaluate the spring back Fig. 7.1.

However, the NDT requires more computational resources to perform a 3D registration in realtime, therefore up to now it has been used in real time only for the map alignment of coarse 2D laser scans (135; 136; 137).

The success of AR-based applications has been driven by the increasing of the performances of personal computers and by the diffusion of several techniques for 3D reconstruction: SLAM with monocular cameras, stereo imaging, structured light, laser range finders, time-of-flight cameras, and range cameras. Structured light is the technology that allows to achieve the higher accuracy of measurement, indeed, dimensions down to $50\text{ }\mu\text{m}$ can be detected (138; 139; 140; 141). This technique is however expensive, the acquisitions tend to be slow, and the observed volume is usually small. Range cameras instead, are particularly cheap devices that have been developed for gaming consoles, and that have had a massive diffusion within research groups thanks to their high quality-cost ratio. The most famous 3D sensors are: the Microsoft Kinect[®], the SoftKinect DS325, and the Asus Xtion[®] (Tab. 7.1). The accuracy of these sensors is not outstanding, nevertheless it is sufficient for numerous applications involving rela-

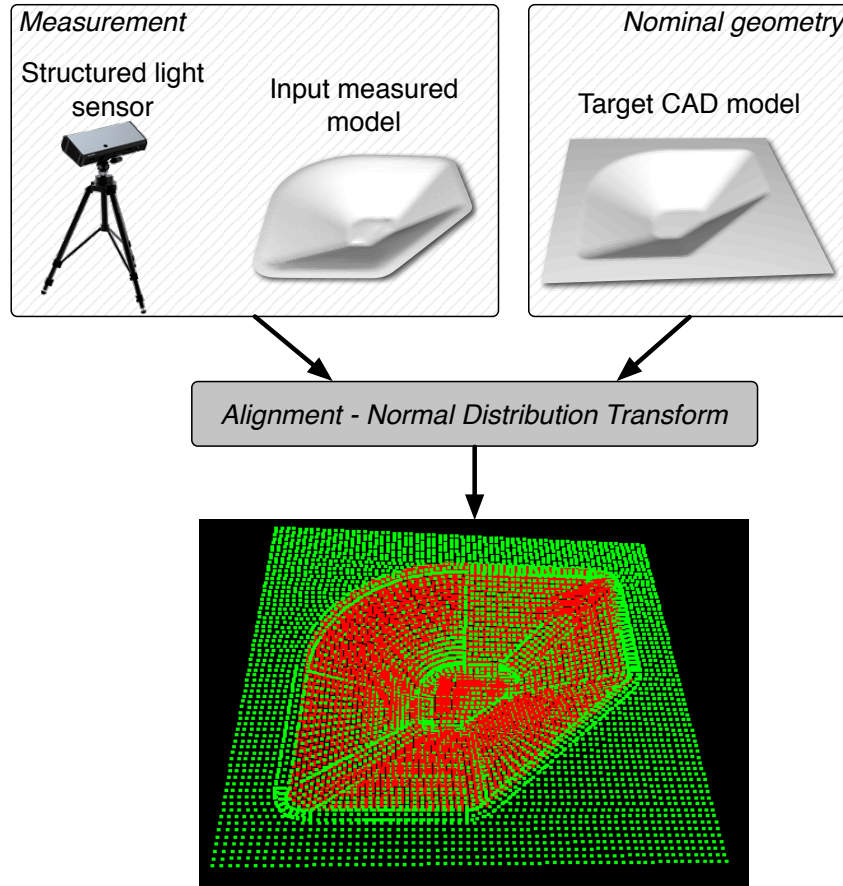


Figure 7.1: Normal Distribution Transform algorithm applied to the alignment of two 3D geometries not explicitly correspondent.

tively large objects (i.e. human tracking and robot navigation).

Moreover, the availability of open source libraries for 3D scene reconstruction is reducing the efforts and the time required to develop robust applications. The most famous and robust registration libraries are the *Kinfu*¹ developed by Newcombe *et al.* (142), and *ReconstructMe*² from Held *et al.* (131). These libraries exploit Graphic Processing Units (GPU) to parallelise the calculations, and reconstruct in real-time (30 fps) a dense 3D map of the environment. Such approach has been exploited in many 3D

¹http://pointclouds.org/documentation/tutorials/using_kinfu_large_scale.php

²<http://reconstructme.net/>

Table 7.1: Examples of 3D sensors available to the market.

Data	Kicrosoft	SoftKinect	Asus Xtion
Data	Kinect	DS325	Pro Live
<i>Depth Camera Resolution</i>	640x480	320x240	640x480
<i>RGB Camera Resolution</i>	1280x960	1280x720	1280x1024
<i>Field of view (HxVxD)</i>	43° x 57°	74° x 58° x 87°	58° x 45° x 70°
<i>Frame rate (fps)</i>	30	60	30
<i>Operating range (m)</i>	1.2 - 3.5	0.15 - 1.0	0.8 - 3.5
<i>Depth resolution (mm)</i>	50 (at 4 m)	14 (at 1 m)	50 (at 4 m)

AR applications, like: 3D mapping (143), gaming (130; 144), surveillance (145), and architecture (146).

7.3 Exploitations

The feasibility study for the 3D scene reconstruction of the working volume is carried out by using the 3D depth camera Asus Xtion Pro Live. The conceptual architecture of the system developed in this study for 3D reconstruction and collision avoidance is outlined in Fig. 7.2. It follows the same structure adopted for the EPC, and it is organised in three layers: sensor, perception and cognitive. In the sensor layer, RGB and 3D images (i.e. point clouds) of the working volume are recorded by the depth camera. The point clouds are then registered in real time by exploiting the Graphic Processing Unit of a graphic card supporting the CUDA library³. The graphic card used is the *NVidia GeForce GTx 690*, which has a 3.0 Kepler architecture (147), 3072 cuda cores, and 4 Gb of dedicated ram. Two registration libraries are tested, the Kinfu and the ReconstructMe, both utilising an efficient implementation of the ICP. The Kinfu is available as module of the Point Cloud Library (PCL), and since it is open source, it is

³<https://developer.nvidia.com/what-cuda>

easy to integrate into large applications. Moreover, the Kinfu is capable to reconstruct large spaces, such as interior of houses and offices (148). The ReconstructMe instead is a licensed software that has better registration capabilities, but it is limited to acquire small volumes.

In Fig. 7.3 it is shown the result of a 10s 3D reconstruction of the working volume through the Kinfu. The accuracy of the virtual model is enough to locate all the rigid bodies that the tool could hit when an erroneous G-code is programmed. Workpiece, vise, and cutting table are clearly identifiable. It is possible to conclude that 3D sensors are suitable for the generation of accurate maps of the working volume.

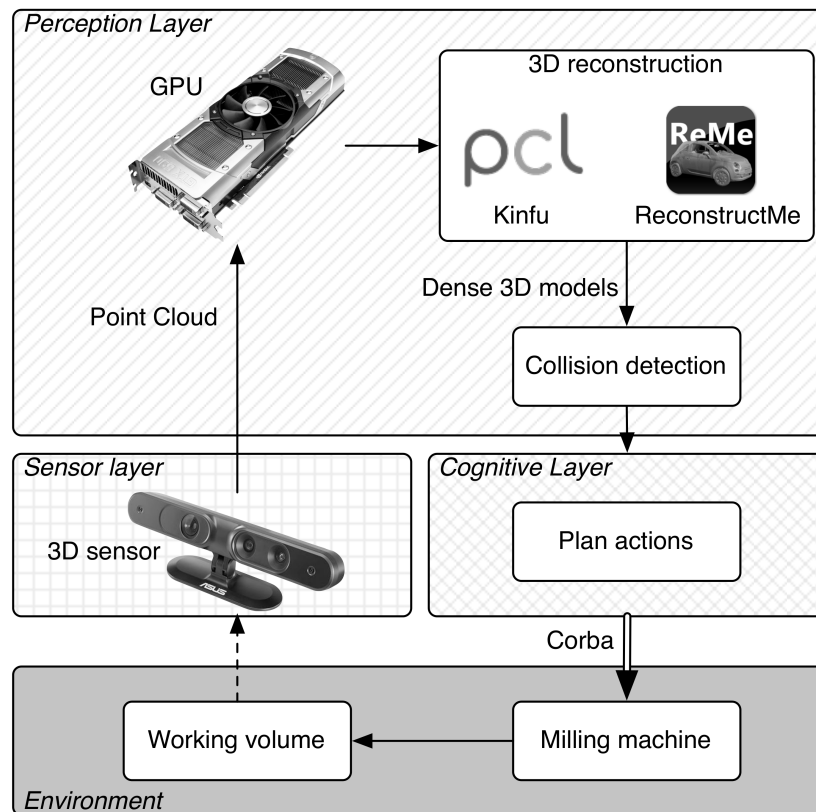


Figure 7.2: Conceptual architecture of the system for the collision avoidance.

In future works these maps can then be utilised, as instance, for collision detection and avoidance. The objects in the acquired scene, might be classified in three categories:

1. Static, constituted by all the non-moving parts.
2. Dynamic, comprising all the moving components.
3. Noise, due not only the measurement uncertainty, but also to the debris and chips produced during the cut.

The moving objects might be then tracked through particle filter, and verified against the global model by collision detection algorithms⁴ such as *CULLIDE* (149) or *RAPID* (150). Whether the distance between objects falls down to a predetermined threshold, or a collision is detected, the control loop is closed by acting on the CNC to stop the axes.

Exploitation of the 3D models acquired could be also the enhancement of the information provided to the CNC user. The Augmented Reality would be used to ease the

⁴A collection of collision detection algorithms is available at <http://gamma.cs.unc.edu/research/collision/packages.html>

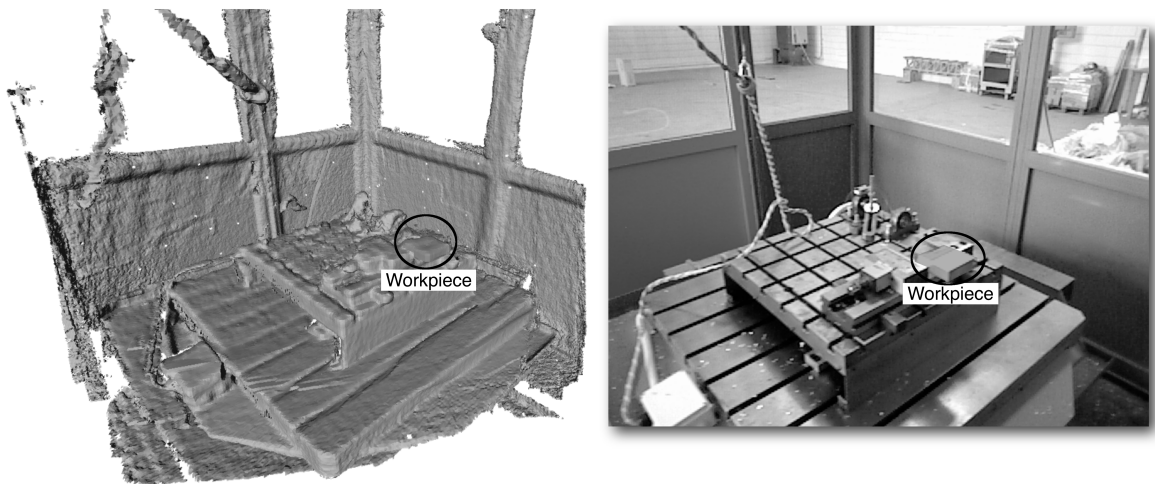


Figure 7.3: Scene reconstructed by using the Kinfu library (left) and RGB image of the working volume (right).

supervision of the process, Fig. 7.4. On the video stream of the operation it might be visualised auxiliary data, such as: process parameters, tool trajectory, tool life, and measured forces or vibrations. It could be possible to superimpose a coloured patch on the tool, thus emphasising the regions of the cut in which the cutter are more stressed, or in which there is the risk of having a collision. The instantaneous shape of the cut, which is usually occluded by the mill and evacuated chip, could also be visualised.

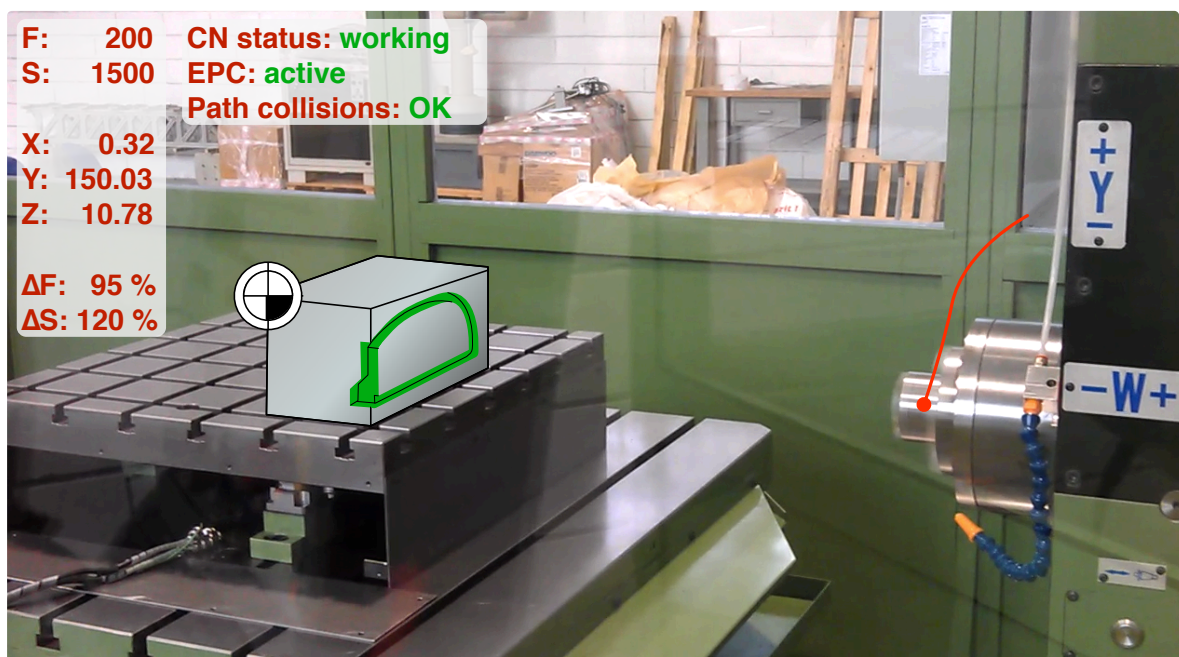


Figure 7.4: Conceptual design of an augmented interface for milling machines.

Chapter 8

Results

In this chapter are summarised the results obtained from the validation of the EPC on real processes. The controller is tested on several tool paths, with complicate geometries. The raw workpiece is made of AA6082-T6 and it has cubic shape in order to ease the definition of its initial geometry in the process simulator. This simplification is not limited by the geometric kernel used to simulate the cut, but eases the setup of the EPC. The ACIS libraries can however handle complicate geometries generated through CAD softwares. The technical specifications of the used tool are reported in Tab. 8.1.

Table 8.1: Data of the utilised tool.

Parameter	Value
Diameter of the mill	63 mm
Number of flutes	5
Insert	WIDIA XPHT160408-AL
Coating of the inserts	TiN

8.1 Offline optimisation

In the tests here described the optimisation is performed offline, that is before the process starts. The target function is defined as the sum of terms considering: process productivity J_p , deflection of the tool $J_{q,p}$, and cutting costs J_c . Forced and self-excited vibrations of the machine are neglected. This simplification is suitable for roughing operations, where the quality of the cut surface is not a mandatory requirement, but the process productivity and tool life have to be maximised. The weights of all the terms of the target function are set to unitary values.

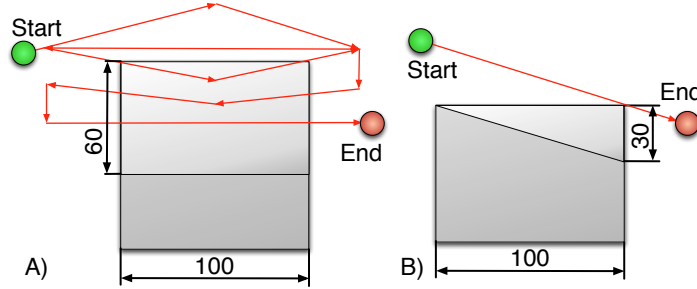


Figure 8.1: Tool paths tested for the offline optimisation.

Two G-codes are programmed: a face milling with a complicate toolpath and cutting depth of 1 mm (Fig. 8.1-A), and a peripheral milling with a cutting depth of 3 mm (Fig. 8.1-B). In both cases, the optimised process is compared with the nominal process. The latter is designed according to the data specified by the tool manufacturer, that is by setting the feed rate to 1000 mm/min , and the spindle speed to 1500 rpm.

The nominal and optimised controls for each operation are reported in Fig. 8.2 and 8.3. In blue are plot the measured feed rate and spindle speed. The solutions calculated by the EPC are plot with a green line. The black line called *set* indicates the instantaneous controls passed by the EPC to the CN. Note that these values are anticipated of 180 ms with respect to the controls computed offline, in order to compensate the latencies

Table 8.2: Cutting times achieved with and without the EPC.

Test	without EPC	with EPC	Time saving
Face milling	82.5 s	44.1 s	- 46.6 %
Peripheral milling	23.3 s	19.5 s	- 22.9 %

associated to the communication and actuation of the drives. By receiving the optimal controls in advance, the machine can follow correctly the desired profile of feed rate and spindle speed. The controls are as much in phase as much the green and the blue line are overlapped. At the end of each block the *Optimized* feed rate goes instantaneously to zero, thus slowing down the axes at the maximum deceleration. This condition has been imposed to speed up the convergence of the OCP, and does not affect the performance of the real process.

For both case studies, the EPC increases the feed rate and reduces the spindle speed when the mill is not cutting (i.e. the cross section is null). As result, the process productivity is maximised and the energy consumption is minimised. On contrary, when the tool is cutting, the feed rate is kept low and the spindle speed is high, thus reducing the force that tends to deflect the tool, and improving the quality of the cut.

The optimal controls allow to reduce the cutting time up to 23 % for the peripheral milling, and up to 46 % for the face milling, Tab. 8.2. However, the improvement of process productivity does not affect the quality of the cut. Roughness measurements are carried out in different zones of the cut surface, as shown in Fig. 8.4 and 8.5. A *Mitutoyo Surftest SJ-401* profilometer is used. The probe has spherical-conical tip with a radius of $2\text{ }\mu\text{m}$. Each coloured segment is 25 mm long and represents the travel of the probe across the surface. For a easier interpretation of the results, the roughness maps of the tests performed with the optimal controls are coloured in green when the quality of the surface is improved, in blue when it remains almost unchanged, and in red when it is worsened. It is concluded that even if the process productivity is significantly improved, the quality of the cut remains overall unchanged.

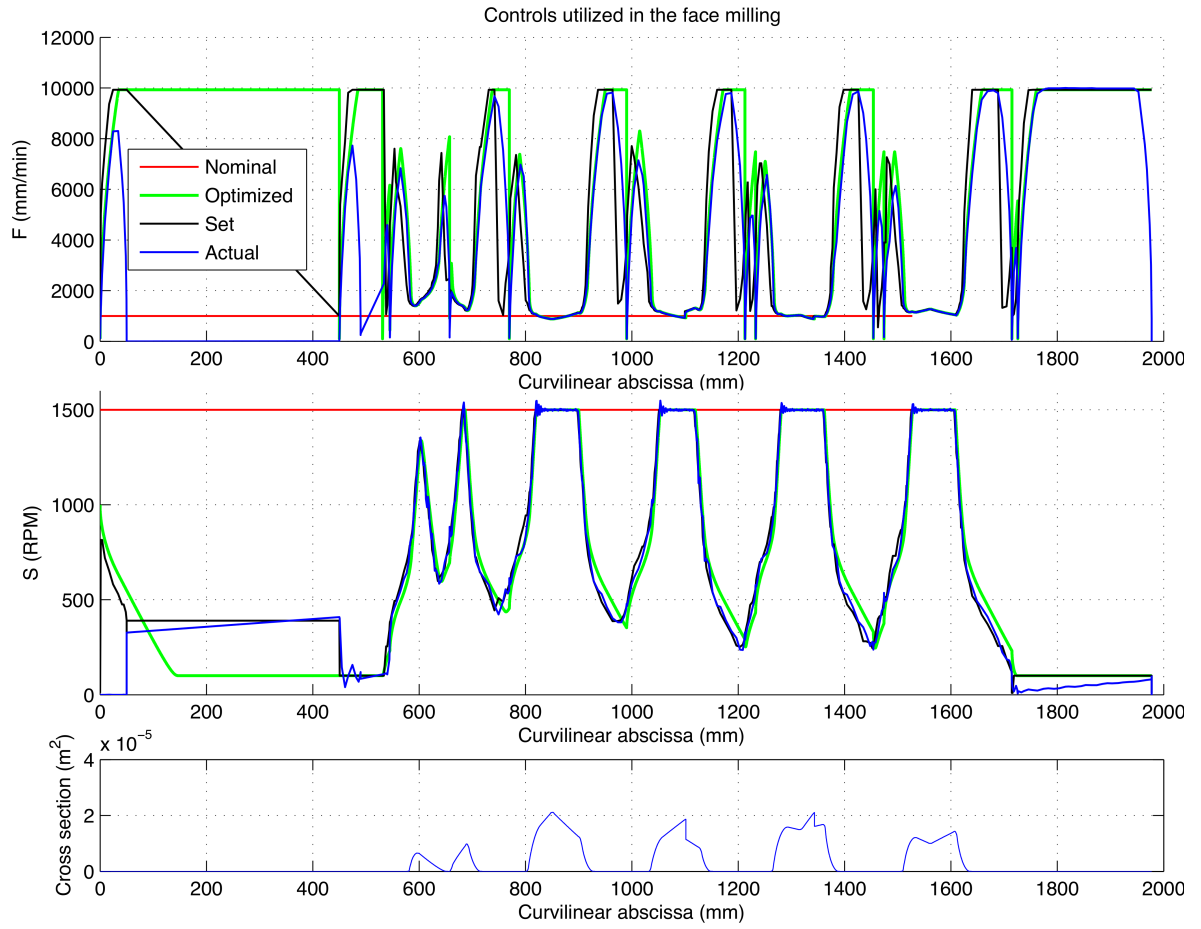


Figure 8.2: Nominal and optimised controls for the face milling.

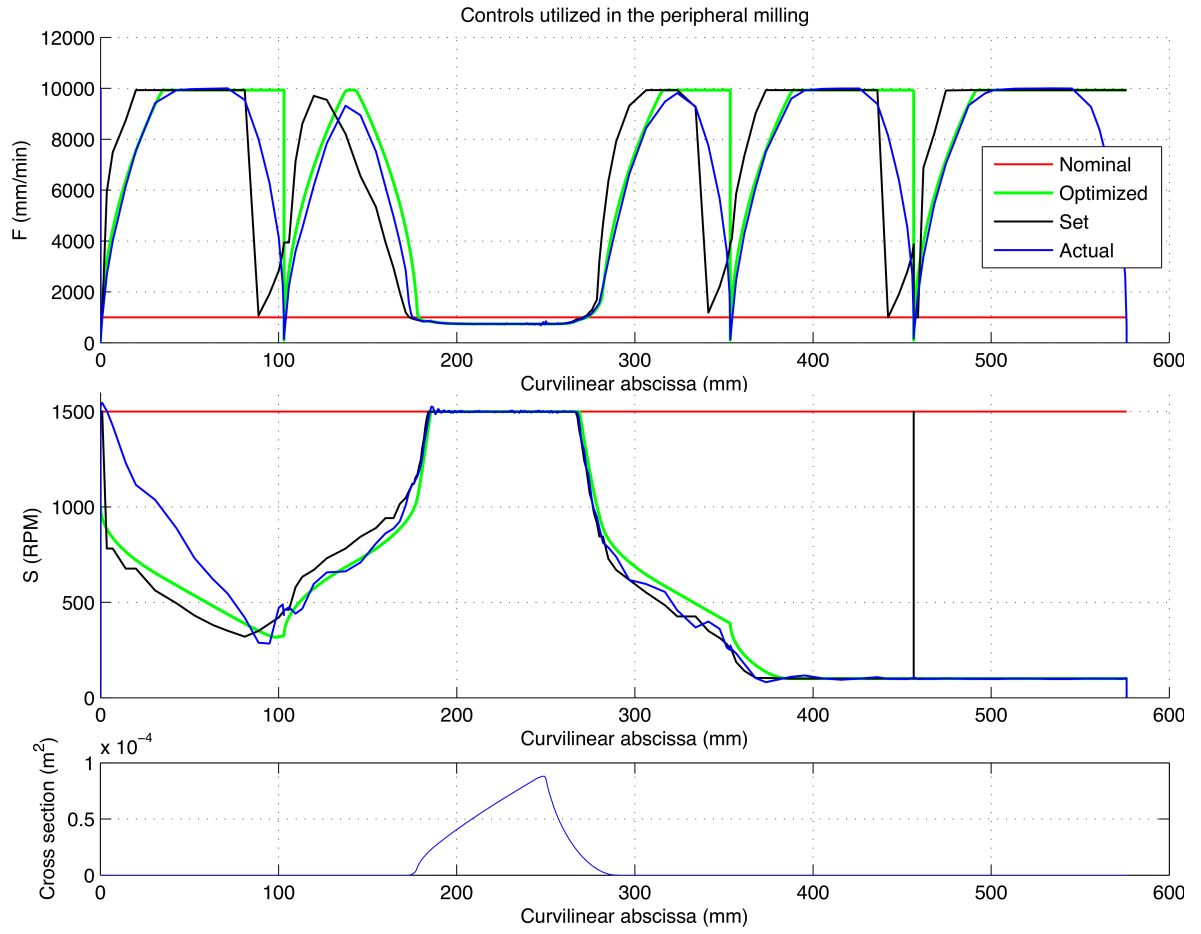


Figure 8.3: Nominal and optimised controls for the peripheral milling.

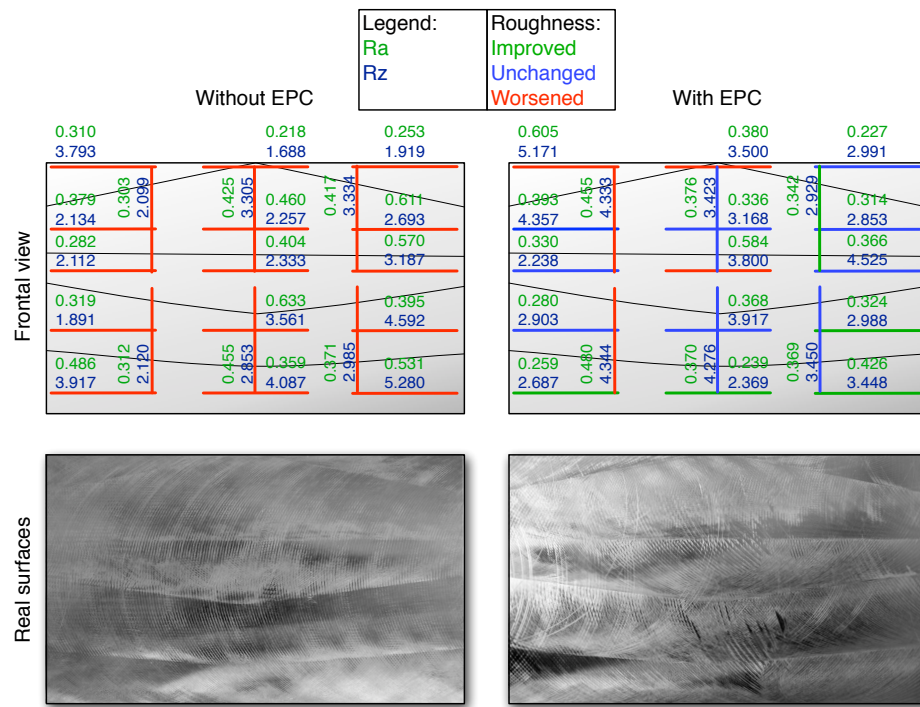


Figure 8.4: Roughness maps for the face milling.

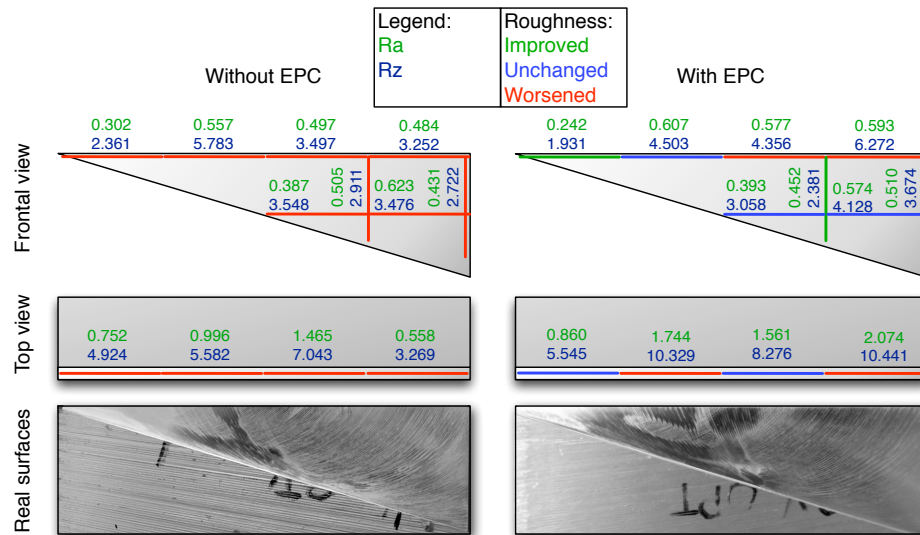


Figure 8.5: Roughness maps for the peripheral milling.

Chapter 9

Conclusion

9.1 Summary of Thesis Achievements

It has been proposed and tested a framework to increase the level of automation of manufacturing processes towards the paradigm of intelligent manufacturing. The developed intelligent system was called Evaluation and Perception Controller, and it was applied to the milling process. The intelligence of the EPC was represented by its capability to enhance the performances of the process, while learning its real behaviour. These features were pursued by developing a modular and flexible architecture, which was constituted by three layers: *sensor*, *perception*, and *cognitive*. The process was sensed through accelerometers, dynamometers, and 3D video cameras displayed in the working volume. Such information was processed by the *perception* layer, which reconstructed the whole state of the system. In the *cognitive* layer the process was simulated to pre-calculate the cut cross section, cutting forces, and stability lobes diagrams. A multi-objective optimisation was then carried out to identify the optimal feed rate and spindle speed.

In order to ease the adaptation of this approach to different manufacturing systems, a language- and OS-independent communication interface between EPC and CNC was defined. Moreover, the experimental tests necessary to calibrate the EPC were described, thus identifying not only the dynamical model of the machine, but also the material properties of the cut workpiece.

The optimisation problem was formulated as an Optimal Control Problem, where a multi objective target function was minimised, while taking into account constraints associated to the dynamic behaviour of the machine and process. The objective function was formulated through terms describing productivity, quality, and costs of the process. These contributes were weighted according to the specific context in which the process was carried out. As instance, for roughing cuts the productivity was weighted more than the remaining terms. During the cut, the models of the process were kept updated by a learning module, thus ensuring the robustness of the OCP.

Finally, the EPC was validated on real processes. Three-axes tool paths were tested, with complicate geometries. The performance of the EPC were validated against nominal processes, designed according to guidelines given by the tool manufacturer. When all the terms of the target function were weighted equally (i.e. for a generic process), the EPC outperformed the productivity achieved in the nominal cuts, without degrading their quality.

9.2 Future Works

This study represents a further step in the direction of developing intelligent manufacturing systems, in which non homogeneous production lines will cooperate as multi-agent robotic system to maximize the performances of production plants. In this vision, a distributed knowledge of the processes involved will be learnt by each single production agent and shared among cloud services to other entities in the work-plant. As

instance, different CNC machine tools will collaborate to define reliable models of the tool wear, or to identify robustly the material properties of the cut workpiece.

In future production plants workmen and process engineers will no more set up processes by identifying the parameters based on their expertise, rather they will specify a production context. Set of requirements will be defined by users, and analysed by intelligent production systems, thus creating a human-machine cooperation in the pursuing of efficiency.

Appendices

Appendix A

Gaussian Processes

A GP is defined as a distribution over functions, and is specified by mean $\mu(x)$ and covariance $k(x, x')$ functions:

$$f \sim \mathcal{GP}(\mu(x), k(x, x')) \quad (\text{A.1})$$

In GP there is not an algebraic formulation of the regressed function, but each regressed point can be thought as a sampled from a multivariate Gaussian distribution. The main advantage of this approach is that the form of the classifier is not limited by a parametric form, instead it can represent a function by letting the data "speak" more clearly for themselves.

Let's consider a data set $x \in \mathbb{R}^n$ with zero mean. Two observations x and x' are related to each other through the *covariance function* $k(x, x')$. Different formulations of $k(x, x')$ can be utilised, to take into account complex relations between points. As instance, it is possible to include constraints in order to model the dynamics of the system (151; 152; 153; 154). The covariance function used to regress the FRFs is

APPENDIX A. GAUSSIAN PROCESSES

formulated as a Squared Exponential function with isotropic distance measure:

$$k(x, x') = \sigma_f e^{-\frac{(x-x')^T P^{-1} (x-x')}{2l}} + \sigma_d^2 \delta_{x,x'} \quad (\text{A.2})$$

where I is the identity matrix, σ_f^2 is the signal variance, l is a scaling factor, σ_d^2 is the variance of the training data, while $\delta_{x,x'}$ is the Kronecker delta function. The second term in Eq. A.2 is included to take into account the noise of measurement of the training data. In order to regress data from the GP, it is necessary to define the likelihood function, which specifies the probability of observations given a latent function. In this study it is used the Gaussian likelihood function since it is the first-choice commonly used in regression:

$$lik(x) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(x-\mu)^T I (x-\mu)}{2\sigma_n^2}} \quad (\text{A.3})$$

μ and σ_n are the mean value and the standard deviation of the likelihood respectively. The former is to set $\mu = 0$ from the assumption made on the null mean. The coefficients l , σ_f , σ_d , σ_n are called *hyperparameters* θ and their values are calculated by training the GP. Given a set of training data (x, f) and a set of test data (x^*, f^*) , the training is carried out by finding hyperparameters that maximise the logarithmic marginal likelihood:

$$\mathcal{L} = \log(y \mid x, \theta) = -\frac{1}{2} \log |\Sigma| - \frac{1}{2} (y - \mu)^T \Sigma^{-1} (y - \mu) - \frac{n}{2} \log(2\pi) \quad (\text{A.4})$$

where y is the output values, and μ is the mean of the inputs, Σ is the covariance of the training data. The first term of \mathcal{L} penalises the complexity of the model, the second is the one that guides the fit of the data, while the third is a normalisation term. Note that the first two terms have opposite effects on the regression: $-\frac{1}{2} \log |\Sigma|$ tends to keep the model simple and general, while $\frac{1}{2} (y - \mu)^T \Sigma^{-1} (y - \mu)$ increases the accuracy of

APPENDIX A. GAUSSIAN PROCESSES

the regression. This definition simplifies the training of the GP since there is no need of a cross validation. \mathcal{L} can be minimised by differentiating the Eq. A.4 with respect to the hyperparameters and solving the associated system of equations.

Up to here it has been summarised how to compute the GP that is used as a *prior* for the Bayesian inference-driven regression. The prior is independent from the training data, but specifies the properties of the function described by the hyperparameters. The joint normal distribution over all the data is defined as:

$$\begin{bmatrix} f \\ f^* \end{bmatrix} = \mathcal{N} \left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma^* \\ \Sigma^{*T} & \Sigma^{**} \end{bmatrix} \right) \quad (\text{A.5})$$

Σ^* is the covariance for the training-test data set, and Σ^{**} is the covariance of test data. The values of f are known, while f^* has to be determined, therefore the conditional distribution of f^* —given f — is:

$$f^* \mid f = \mathcal{N}(\mu^* + \Sigma^{*T} \Sigma^{-1}(f - \mu), \Sigma^{**} - \Sigma^{*T} \Sigma^{-1} \Sigma^*) \quad (\text{A.6})$$

that is the *posterior* distribution for a set of test data x^* . The corresponding posterior process for a set of θ hyperparameters is then defined by the mean μ_θ and variance $k_\theta(x, x')$, as follows:

$$\mu_\theta(x) = \mu(x) + \Sigma(X, x)^T \Sigma^{-1}(f - \mu) \quad (\text{A.7})$$

$$k_\theta(x, x') = k(x, x') - \Sigma(X, x)^T \Sigma^{-1} \Sigma(X, x') \quad (\text{A.8})$$

with $\Sigma(X, x)$ being the covariance calculated between every training case and x . It can be noted that adding a data to the training process enriches the processed information, with a consequent reduction of the posterior variance $k_\theta(x, x')$. Indeed, to the prior variance $k(x, x')$ it is subtracted a positive quantity calculated from the training tests.

Appendix B

Calculation of Stability Lobes diagram

The model here described is the one developed by Altintas and Budak (53). This appendix summarises the basic concepts that allow to compute cutting forces and stability lobes diagrams.

Cutting forces

The model considers two degrees of freedom, which are the directions orthogonal to the axis of the tool, Fig. B.1. It is assumed that the mill has n_t teeth, and a null helix angle. The feed is along X . During the cut, the dynamic displacement of the j -th tooth along the radial direction is given by

$$v_j = -x \sin(\phi_j) - y \cos(\phi_j) \quad (\text{B.1})$$

where ϕ_j is the instantaneous angular immersion. For a tooth rotating with a Ω angular speed, the dynamics of the immersion angle is $\phi_j(t) = \Omega t$. The total chip thickness is given by two contributes, associated to the rigid movement of the cutter, and to the

APPENDIX B. CALCULATION OF STABILITY LOBES DIAGRAM

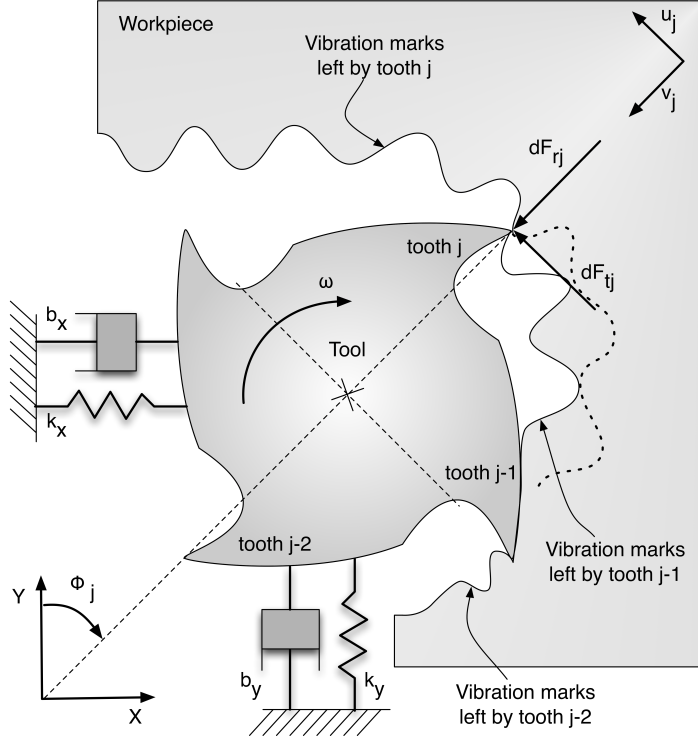


Figure B.1: Two-dimensional dynamical model of milling process.

vibrations of the tool at the present and previous tooth periods:

$$h(\phi_j) = [f_z \sin(\phi_j) + (v_{j,0} - v_j)] g(\phi_j) \quad (B.2)$$

where f_z is the feed per tooth and $(v_j, v_{j,0})$ are the dynamic displacements of the cutter at the current and previous tooth period. The function $g(\phi_j)$ is unitary in the range of immersion angles in which the mill is cutting, that is when ϕ_j is between the entry (ϕ_{st}) and exit (ϕ_{ex}) angles, otherwise it is zero. The static component of the chip thickness does not contribute to the vibrations of the tool, therefore the Eq. B.3 is simplified and by substituting the Eq. B.1 it becomes:

$$h(\phi_j) = [\Delta x \sin(\phi_j) + \Delta y \cos(\phi_j)] g(\phi_j) \quad (B.3)$$

APPENDIX B. CALCULATION OF STABILITY LOBES DIAGRAM

where $\Delta x = x - x_0$ and $\Delta y = y - y_0$. The dynamic displacements of the cutter at the present and previous tooth period are (x, y) and (x_0, y_0) respectively.

The cutting force $F_{c,j}$ on the $X-Y$ plane can be decomposed into its tangential $F_{t,j}$ and radial $F_{r,j}$ components. The latter are proportional to the depth of cut d , chip thickness $h(\phi_j)$, as well as the constants K_t and K_r taking into account the material properties of the workpiece, tool, the geometry of the cutter, and the lubrication conditions:

$$F_{t,j} = K_t d h(\phi_j) , \quad F_{r,j} = K_r F_{t,j} \quad (\text{B.4})$$

The forces can be projected along the X and Y directions and summed among all the teeth, then:

$$\begin{aligned} F_x &= \sum_{j=0}^{n_t-1} [-F_{t,j} \cos(\phi_j) - F_{r,j} \sin(\phi_j)] \\ F_y &= \sum_{j=0}^{n_t-1} [F_{t,j} \sin(\phi_j) - F_{r,j} \cos(\phi_j)] \end{aligned} \quad (\text{B.5})$$

the immersion angle is calculated as $\phi_j = \phi_p + j \phi$, where $\phi_p = 2\pi/n_t$ is the pitch angle. A more practical representation of the cutting forces is given by substituting the Eq. B.3 and Eq. B.4 into Eq. B.5, and rearranging the expression in matrix form it gives:

$$\{F_c(t)\} = \frac{1}{2} d K_t [A(t)] \{\Delta(t)\} \quad (\text{B.6})$$

APPENDIX B. CALCULATION OF STABILITY LOBES DIAGRAM

where $\{F_c(t)\} = (F_x(t), F_y(t))^T$, $\{\Delta(t)\} = (\Delta x, \Delta y)^T$, and $[A(t)]$ is the *directional dynamic milling force matrix* whose components are:

$$\begin{aligned}
 a_{x,x} &= \sum_{j=0}^{n_t-1} -g [\sin(2\phi_j) + K_r (1 - \cos(2\phi_j))] \\
 a_{x,y} &= \sum_{j=0}^{n_t-1} -g [(1 + \cos(2\phi_j)) + K_r \sin(2\phi_j)] \\
 a_{y,x} &= \sum_{j=0}^{n_t-1} -g [(1 - \cos(2\phi_j)) - K_r \sin(2\phi_j)] \\
 a_{y,y} &= \sum_{j=0}^{n_t-1} -g [\sin(2\phi_j) - K_r (1 + \cos(2\phi_j))]
 \end{aligned} \tag{B.7}$$

As the cutter rotates, the milling forces vary periodically. $[A(t)]$ is periodic at tooth passing frequency $\omega = n_t \Omega$ (or tooth period $T = 2\pi/\omega$), therefore it can be expanded into Fourier series:

$$[A(t)] = \sum_{r=-\infty}^{\infty} [A(t)] e^{ir\omega t} [A_r] = \frac{1}{T} \int_0^T [A(t)] e^{-ir\omega t} dt \tag{B.8}$$

r is the number of harmonics to be considered in the approximation of $[A(t)]$ and depends on the immersion conditions and on the number of teeth in cut. In the most simple model $r = 0$ and only the average component of $[A(t)]$ is considered:

$$[A_0] = \frac{1}{T} \int_0^T [A(t)] dt \tag{B.9}$$

APPENDIX B. CALCULATION OF STABILITY LOBES DIAGRAM

Since $[A_0]$ is valid only when the mill is cutting (i.e. $\phi_{st} \leq \phi \leq \phi_{ex}$), it is equal to the average value of $[A(t)]$ at $\phi = \phi_p$:

$$[A_0] = \frac{1}{\phi_p} \int_{\phi_{st}}^{\phi_{ex}} [A(\phi)] d\phi = \frac{n_t}{2\pi} \begin{bmatrix} \alpha_{x,x} & \alpha_{x,y} \\ \alpha_{y,x} & \alpha_{y,y} \end{bmatrix} \quad (\text{B.10})$$

where:

$$\begin{aligned} \alpha_{x,x} &= \frac{1}{2} [\cos(2\phi) - 2K_r \phi + K_r \sin(2\phi)]_{\phi_{st}}^{\phi_{ex}} \\ \alpha_{x,y} &= \frac{1}{2} [-\sin(2\phi) - 2\phi + K_r \cos(2\phi)]_{\phi_{st}}^{\phi_{ex}} \\ \alpha_{y,x} &= \frac{1}{2} [-\sin(2\phi) + 2\phi + K_r \cos(2\phi)]_{\phi_{st}}^{\phi_{ex}} \\ \alpha_{y,y} &= \frac{1}{2} [-\cos(2\phi) - 2K_r \phi - K_r \sin(2\phi)]_{\phi_{st}}^{\phi_{ex}} \end{aligned} \quad (\text{B.11})$$

and the dynamic expression of the cutting force in Eq. B.6 becomes:

$$\{F_c(t)\} = \frac{1}{2} d K_t [A_0] \{\Delta(t)\} \quad (\text{B.12})$$

note that $[A_0]$ is time invariant, but it depends on the immersion angle which, in general, varies in time.

Stability Lobes Diagrams

The transfer function $[G(i\omega)]$ experimentally identified at the cutter-workpiece contact zone is represented as a 2×2 matrix:

$$[G(i\omega)] = \begin{bmatrix} G_{x,x}(i\omega) & G_{x,y}(i\omega) \\ G_{y,x}(i\omega) & G_{y,y}(i\omega) \end{bmatrix} \quad (\text{B.13})$$

APPENDIX B. CALCULATION OF STABILITY LOBES DIAGRAM

where $G_{x,x}(i\omega)$ and $G_{y,y}(i\omega)$ are the direct transfer functions in x and y directions, and $G_{x,y}(i\omega)$ and $G_{y,x}(i\omega)$ are the cross transfer functions. The vibrations vectors at present time and previous tooth period T are called:

$$\{r\} = \{x(t), y(t)\}^T ; \quad \{r_0\} = \{x(t-T), y(t-T)\}^T$$

which, at the chatter frequency ω_c , can be calculated as:

$$\begin{aligned} \{r(i\omega_c)\} &= [G(i\omega)] \{F\} e^{i\omega_c t} \\ \{r_0(i\omega_c)\} &= \{r(i\omega_c)\} e^{-i\omega_c T} \end{aligned} \tag{B.14}$$

The displacement $\{\Delta\} = \{(x-x_0)(y-y_0)\}^T$ is then given by:

$$\begin{aligned} \{\Delta(i\omega_c)\} &= \{r(i\omega_c)\} - \{r_0(i\omega_c)\} \\ &= [1 - e^{-i\omega_c T}] e^{i\omega_c t} [G(i\omega_c)] \{F\} \end{aligned} \tag{B.15}$$

note that $\omega_c T$ is the phase delay between the vibrations at successive tooth periods T . Substituting Eq. B.15 into Eq. B.12 defined to calculate the cutting force, it is obtained the equation:

$$\{F\} e^{-i\omega_c t} = \frac{1}{2} d K_t [1 - e^{-i\omega_c T}] [A_0] [G(i\omega_c)] \{F\} e^{-i\omega_c t}$$

whose non trivial solution is given by the characteristic equation of the closed loop dynamical milling system:

$$\det \left[[I] - \frac{1}{2} d K_t (1 - e^{-i\omega_c T}) [A_0] [G(i\omega_c)] \right] = 0 \tag{B.16}$$

The eigenvalues Λ of the characteristic equation are calculated after the chatter frequency, material properties (K_t, K_r), radial immersion, and transfer function of the structure have been defined. Since $[G(i\omega)]$ is matrix of complex number, the eigenval-

APPENDIX B. CALCULATION OF STABILITY LOBES DIAGRAM

ues have a real and an imaginary part: $\Lambda = \Lambda_R + i \Lambda_I$. After some simple algebraic calculation steps here neglected for brevity, it is possible to obtain the critical axial depth of cut at $\omega = \omega_c$:

$$d_{lim} = -\frac{2\pi \Lambda_R}{n_t K_t} (1 + \kappa^2) \quad (\text{B.17})$$

where $\kappa = \Lambda_I/\Lambda_R$. Given a chatter frequency, the Eq. B.17 allows to calculate the maximum cutting depth that gives a stable process.

The stable spindle speed S is calculated by estimating the tooth passing period T , given by:

$$\begin{aligned} T &= \frac{1}{\omega_c} (\epsilon + 2 k \pi) \\ S &= \frac{60}{n_t T} \end{aligned} \quad (\text{B.18})$$

where k is the integer number of the lobes considered, and $\epsilon = \pi - 2(\tan^{-1}(k))$ is the phase shift between present and previous vibration marks.

Appendix C

Normal Distribution Transform

The NDT is a registration algorithm that has been developed to align two different sets of 2D or 3D points. Each set is called *point cloud*. Here it is described the NDT implementation in the open source Point Cloud Library (PCL), (155).

First, the input cloud P_{input} (i.e. the cloud to be aligned) is down-sampled, in order to increase the speed of the registration. This operation is performed through a voxel grid filter. A voxel is a 3D cube, that groups N samples and describes the associated space through their centroid. The reduced cloud is called \hat{P}_{input} . The NDT algorithm is then outlined in two steps: the creation of a statistical model of the 3D target cloud P_{target} , and the probability-driven registration of P_{input} .

Statistical model

The P_{target} is subdivided into cells with constant size and containing at least three samples. The size of the cells is determined according to the dimension of the point

APPENDIX C. NORMAL DISTRIBUTION TRANSFORM

cloud. In general a trade-off should be sought, since alignments performed with small cells are more precise, but require more computational resources. For a k -th cell, its centroid $q_{\text{target},k}$ and covariance matrix $\Sigma_{\text{target},k}$ is calculated as:

$$q_{\text{target},k} = \frac{1}{n_p} \sum_{j=1}^{n_p} P_{\text{target},j} \quad (\text{C.1})$$

$$\Sigma_{\text{target},k} = \frac{1}{n_p} \sum_{j=1}^{n_p} (P_{\text{target},j} - q) (P_{\text{target},j} - q)^T \quad (\text{C.2})$$

where n_p is the number of points in the cell. A normal probability distribution $N(q, \Sigma)$ is associated to each cell, thus obtaining a global statistical model of the 3D cloud P_{target} . Since a derivative optimisation algorithm is used to minimise the alignment error, the global statistical model is smoothed by partially overlapping the local normal distributions.

Given a point P , the probability that it belongs to the global probability distribution p is then calculated as:

$$p(P) \sim \exp \left(-\frac{(P - q)^T \Sigma^{-1} (P - q)}{2} \right) \quad (\text{C.3})$$

Alignment

The roto-translation matrix used to align the input cloud to the target cloud can be represented by a vector x of six components: three for the rotations, and three for the translations. These elements are identified through Newton's optimisation algorithm. The initial transformation guess x_o is set equal to the identity matrix, thus assuming that initially the two clouds are already aligned. The first guess is then corrected according to the value of a score function that is calculated by iterating across the following steps:

APPENDIX C. NORMAL DISTRIBUTION TRANSFORM

- project \hat{P}_{input} into the frame of P_{target} according to the calculated transformation x . The mapped input cloud is called \hat{M}_{input} .
- compute the corresponding normal distribution for each $\hat{M}_{\text{input},j}$ point, with $j = 1, \dots, n_p$.
- evaluate the distribution for each $\hat{M}_{\text{input},j}$ point, and sum the result, thus obtaining the score:

$$\text{score}(x) = - \sum_{j=1}^{n_i} \exp \left(\frac{\left(\hat{M}_{\text{input},j} - q_j \right)^T \Sigma_j^{-1} \left(\hat{M}_{\text{input},j} - q_j \right)}{2} \right) \quad (\text{C.4})$$

where n_i is the number of points in the input cloud, while q_i and Σ_j are the mean and covariance matrix of the normal distribution corresponding to $\hat{M}_{\text{input},j}$, looked up in the NDT of the target cloud.

The convergence criteria can be established based on the maximum number of iterations, as well as on the minimum step of transformation.

Appendix D

Identification of the cutting constants of the workpiece

In this study the reference material of the workpiece is the *AA 6082-T6*. To compute the lobes diagram and cutting force it is necessary to estimate the cutting constants of the material through experimental tests. It is followed the procedure developed by Budak *et al.* (156). Full-immersion face milling (i.e. slotting) cuts are carried at different feed rates. The used tool, inserts, and spindle speed are listed in Tab. D.1.

Cutting forces along the three cartesian axes of the machine (\bar{F}_x , \bar{F}_z , and \bar{F}_y) are measured with a dynamometric table. The average cutting forces at each feed rate are reported in Fig. D.1. The relation between cutting forces and cutting force constants

Table D.1: Process parameters used for the cutting force constants identification.

Parameter	Value
Material of the workpiece	AA 6082-T6
Diameter of the mill	63 mm
Number of flutes	5
Insert	WIDIA XPHT160408-AL
Coating of the inserts	TiN
Spindle speed	1500 rpm
Depth of cut	2 mm
Width of cut	63 mm

APPENDIX D. IDENTIFICATION OF THE CUTTING CONSTANTS OF THE WORKPIECE

is linear, and it is defined as:

$$\begin{cases} \bar{F}_x = -\frac{n_t d}{4} K_{rc} f - \frac{n_t d}{4} K_{re} \\ \bar{F}_y = +\frac{n_t d}{4} K_{tc} f + \frac{n_t d}{4} K_{te} \\ \bar{F}_z = +\frac{n_t d}{4} K_{ac} f + \frac{n_t d}{4} K_{ae} \end{cases} \quad (D.1)$$

where f is the feed rate n_t is the number of flutes in the mill, and d is the cutting depth. K_{rc} , K_{tc} , and K_{ac} are the cutting constants due to the shear stresses in radial, tangential, and axial directions respectively, while K_{re} , K_{te} , and K_{ae} are due to the frictions between the edge of the tool and the workpiece. The values of the coefficients are reported in Tab. D.2.

The axial component of the cutting force, that is F_z , is small and it is almost constant with respect to feed rate. Therefore K_{ac} is considered to be 0 N/mm^2 . Such assumption does not affect the calculation of the lobes diagram, since the model used in this study considers only the forces perpendicular to the axis of the spindle.

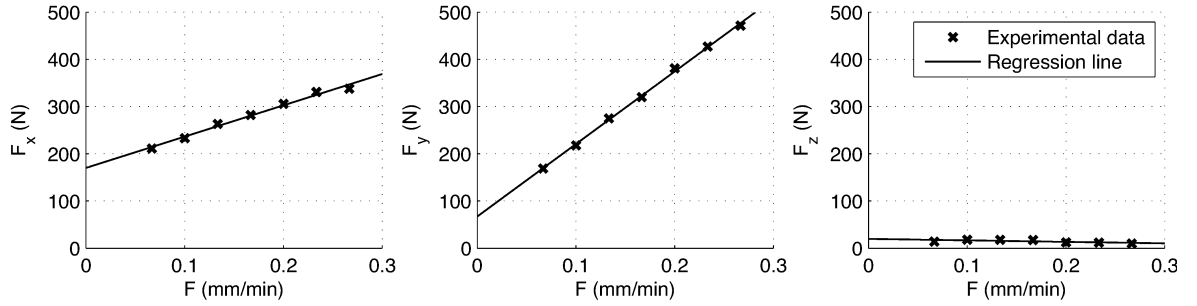


Figure D.1: Cutting forces at different feed rates. The components of the force are parallel (left) to the feed, perpendicular (centre), and along the axis of the tool (right).

Table D.2: Cutting force constants for the AA 6082-T6.

Constant	K_{rc}	K_{re}	K_{tc}	K_{te}	K_{ac}	K_{ae}
Value	264.9 N/mm^2	53.4 N/mm	614.1 N/mm^2	21.1 N/mm	0 N/mm^2	3.9 N/mm

Appendix E

Technical drawings of the Alesamonti MB63

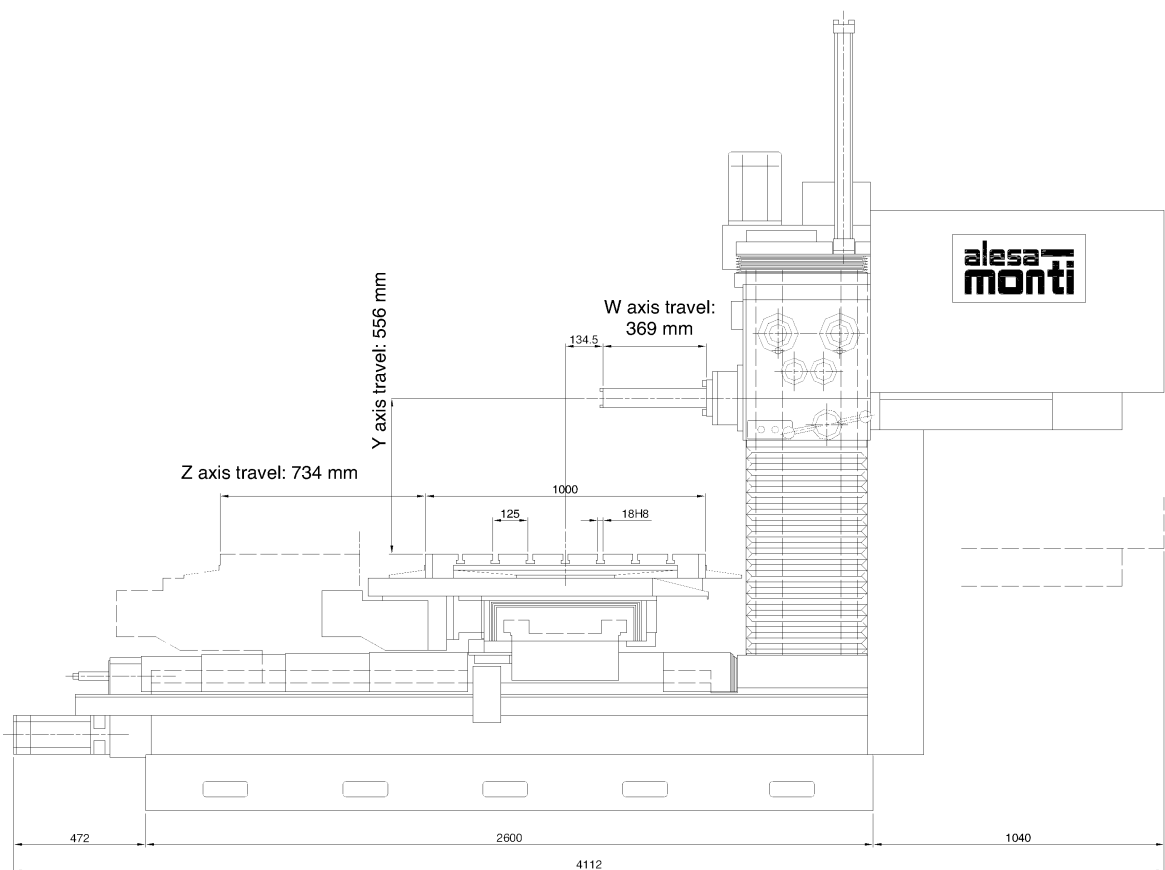


Figure E.1: Lateral view.

APPENDIX E. TECHNICAL DRAWINGS OF THE ALESAMONTI
MB63

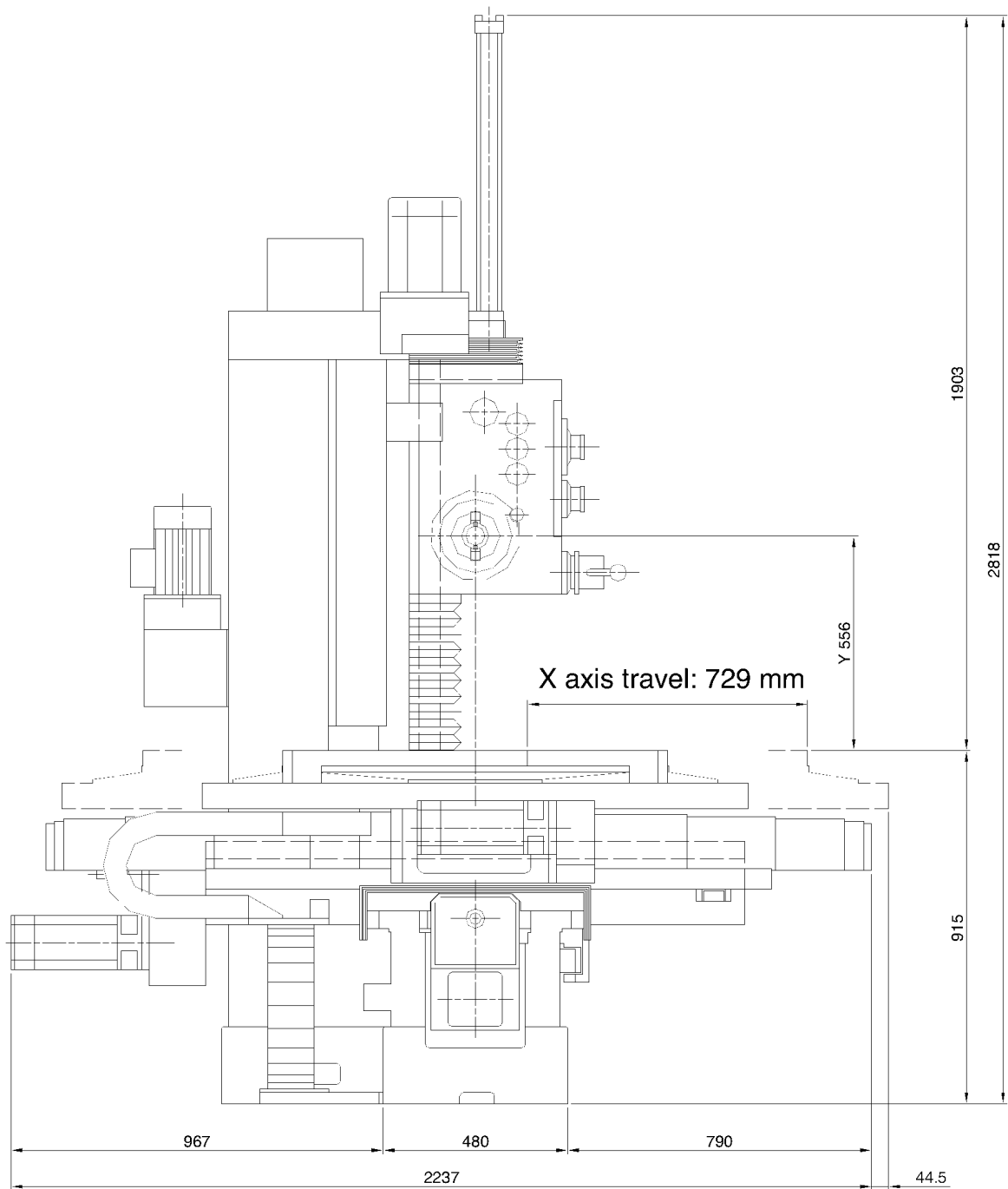


Figure E.2: Frontal view.

Appendix F

Information logged by the EPC

Table F.1: Codes of the machine status parameter.

Code	Meaning
0	Manual mode.
1	Automatic mode.
2	Block syntax error.
3	Waiting to start.
5	Stop.
6	Executing G-code program or block.

APPENDIX F. INFORMATION LOGGED BY THE EPC

Table F.2: Structure of the Status C-struct.

Data	Field	Unit of meas.	Description
<i>Time</i>	t	ms	Time in milliseconds elapsed since the milling process has started.
<i>Curvilinear abscissa</i>	blkspace	mm	Travelled length since the last block has started.
<i>Axes positions</i>	xm	mm	Position of the five axis referred to the local user-defined workpiece frame.
	ym	double	
	zm	double	
	wm	double	
	bm	double	
<i>Axes speed</i>	xv	mm/min	Mean axis speed calculated starting from the values transduced by the main counting device. <i>_max</i> and <i>_min</i> data available.
	yv	double	
	zv	double	
	wv	double	
	bv	double	
<i>Feed rate</i>	F	mm/min	Programmed feed rate, upstream of the overrides. <i>_max</i> and <i>_min</i> data available.
	F_o	mm/min	Programmed feed rate, downstream of the overrides. <i>_max</i> and <i>_min</i> data available.
	F_instvel	mm/min	Instantaneous feed rate.
<i>Spindle speed</i>	S	RPM	Programmed spindle speed, upstream of the overrides. <i>_max</i> and <i>_min</i> data available.
	S_o	RPM	Programmed spindle speed, downstream of the overrides. <i>_max</i> and <i>_min</i> data available.
	S_instvel	RPM	Instantaneous spindle speed.
<i>Absorbed current</i>	current	%	Absorbed current by the spindle. <i>_max</i> and <i>_min</i> data available.
<i>Absorbed power</i>	power	W	Absorbed electrical power by the spindle. <i>_max</i> and <i>_min</i> data available.
<i>Torque at the spindle</i>	torque	Nm	Torque at the spindle. <i>_max</i> and <i>_min</i> data available.
<i>Machine status</i>	jobstatus	/	Code used to identify the machine status. See table F.1
<i>Block number</i>	nblk	/	Number of the executed block.
<i>Active block</i>	blk	/	String with the block currently active.
<i>Executed file</i>	iso_file	/	String with the executed part program.

APPENDIX F. INFORMATION LOGGED BY THE EPC

Table F.3: List of parameters and sub parameters listened by the Listener instance.

M.XAC		
Sub parameter	Unit of Meas.	Note
<i>XM, YM, ZM, WM, BM</i>	<i>mm</i>	Axis positions.
MIC_BLK_F_INFO		
Sub parameter	Unit of Meas.	Note
<i>ACTBLK</i>	/	String of the executed block.
<i>ACTFILE</i>	/	Name of the executed program.
MIC_INST_DATA		
Sub parameter	Unit of Meas.	Note
<i>MS.TIME</i>	<i>ms</i>	Time elapsed since the process started.
<i>MXA.X</i>	<i>mm</i>	Axes position in the machine frame.
<i>MXA.Y</i>	<i>mm</i>	
<i>MXA.Z</i>	<i>mm</i>	Curvilinear abscissae of the active block.
<i>MXA.A1</i>	<i>mm</i>	
<i>MXA.A2</i>	<i>deg</i>	Machine state, see Tab. F.1
<i>BLKSPACE</i>	<i>mm</i>	
<i>JOBSTATUS</i>	/	Average axis speed.
<i>INSTV.X_AVG</i>	<i>mm/min</i>	
<i>INSTV.Y_AVG</i>	<i>mm/min</i>	Feed rate upstream the override.
<i>INSTV.Z_AVG</i>	<i>mm/min</i>	
<i>INSTV.A1_AVG</i>	<i>mm/min</i>	Feed rate downstream the override.
<i>INSTV.A2_AVG</i>	<i>mm/min</i>	
<i>INSTV.X_MIN</i>	<i>mm/min</i>	Spindle speed upstream the override.
<i>INSTV.Y_MIN</i>	<i>mm/min</i>	
<i>INSTV.Z_MIN</i>	<i>mm/min</i>	Spindle speed downstream the override.
<i>INSTV.A1_MIN</i>	<i>mm/min</i>	
<i>INSTV.A2_MIN</i>	<i>mm/min</i>	Absorbed current.
<i>INSTV.X_MAX</i>	<i>mm/min</i>	
<i>INSTV.Y_MAX</i>	<i>mm/min</i>	Absorbed electrical power.
<i>INSTV.Z_MAX</i>	<i>mm/min</i>	
<i>INSTV.A1_MAX</i>	<i>mm/min</i>	Torque at the spindle.
<i>INSTV.A2_MAX</i>	<i>mm/min</i>	
<i>F.NOV_AVG</i>	<i>mm/min</i>	Torque at the spindle.
<i>F.NOV_MIN</i>	<i>mm/min</i>	
<i>F.NOV_MAX</i>	<i>mm/min</i>	Torque at the spindle.
<i>F.OV_AVG</i>	<i>mm/min</i>	
<i>F.OV_MIN</i>	<i>mm/min</i>	Torque at the spindle.
<i>F.OV_MAX</i>	<i>mm/min</i>	
<i>S.NOV_AVG</i>	<i>mm/min</i>	Torque at the spindle.
<i>S.NOV_MIN</i>	<i>mm/min</i>	
<i>S.NOV_MAX</i>	<i>mm/min</i>	Torque at the spindle.
<i>S.OV_AVG</i>	<i>mm/min</i>	
<i>S.OV_MIN</i>	<i>mm/min</i>	Torque at the spindle.
<i>S.OV_MAX</i>	<i>mm/min</i>	
<i>SPD.CURRENT_AVG</i>	1000 = 100 %	Torque at the spindle.
<i>SPD.CURRENT_MIN</i>	1000 = 100 %	
<i>SPD.CURRENT_MAX</i>	1000 = 100 %	Torque at the spindle.
<i>SPD.POWER_AVG</i>	<i>W</i>	
<i>SPD.POWER_MIN</i>	<i>W</i>	Torque at the spindle.
<i>SPD.POWER_MAX</i>	<i>W</i>	
<i>SPD.TORQUE_AVG</i>	<i>Nm</i>	Torque at the spindle.
<i>SPD.TORQUE_MIN</i>	<i>Nm</i>	
<i>SPD.TORQUE_MAX</i>	<i>Nm</i>	

Bibliography

- [1] S. Kalpakjian, *Manufacturing engineering and technology*. World Student Series, Addison-Wesley Pub. Co., 1992.
- [2] G. Amber and P. Amber, *Anatomy of automation*. New York, NY, USA: Prentice-Hall, 1st ed., 1962.
- [3] Y. Koren, *Computer Control of Manufacturing Systems*. New York, NY, USA: McGraw-Hill, Inc., 1st ed., 1983.
- [4] G. Weiss, ed., *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge, MA, USA: MIT Press, 1999.
- [5] S. Feng, K. Stouffer, and K. Jurrens, “Manufacturing planning and predictive process models integration using software agents,” *Advanced Engineering Informatics*, vol. 19, pp. 135–142, 2005.
- [6] R. Teti, K. Jemielniak, G. OâDonnell, and D. Dornfeld, “Advanced monitoring of machining operations,” *{CIRP} Annals - Manufacturing Technology*, vol. 59, no. 2, pp. 717 – 739, 2010.
- [7] A. Nee, S. Ong, G. Chryssolouris, and D. Mourtzis, “Augmented reality applications in design and manufacturing,” *{CIRP} Annals - Manufacturing Technology*, vol. 61, no. 2, pp. 657 – 679, 2012.

- [8] M. C. Kang, J. S. Kim, and J. H. Kim, “A monitoring technique using a multi-sensor in high speed machining,” *Journal of Materials Processing Technology*, vol. 113, no. 1â3, pp. 331 – 336, 2001. jce:titlej5th Asia Pacific conference on Materials processingj/ce:titlej.
- [9] T. Kalvoda and Y.-R. Hwang, “Analysis of signals for monitoring of nonlinear and non-stationary machining processes,” *Sensors and Actuators A: Physical*, vol. 161, no. 1â2, pp. 39 – 45, 2010.
- [10] G. Franklin, J. Powell, and M. Workman, *Digital control of dynamic systems*. Addison-Wesley world student series, Addison Wesley Longman, 1998.
- [11] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House radar library, Artech House, 2004.
- [12] A. Bry, A. Bachrach, and N. Roy, “State estimation for aggressive flight in gps-denied environments using onboard sensing,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1–8, 2012.
- [13] G. Rigatos, “Nonlinear kalman filters and particle filters for integrated navigation of unmanned aerial vehicles,” *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 978 – 995, 2012.
- [14] G. Rigatos, “Extended kalman and particle filtering for sensor fusion in motion control of mobile robots,” *Mathematics and Computers in Simulation*, vol. 81, no. 3, pp. 590 – 607, 2010. jce:titlejThe Sixth {IMACS} Seminar on Monte Carlo Methodsj/ce:titlej jce:titlejApplied Scientific Computing VII. Forward Numerical Grid Generation, Approximation and Simulationj/ce:titlej.
- [15] X. Chen, H. Gong, and J. Wang, “{BRT} vehicle travel time prediction based on

- {SVM} and kalman filter,” *Journal of Transportation Systems Engineering and Information Technology*, vol. 12, no. 4, pp. 29 – 34, 2012.
- [16] P. Biber and W. Strasser, “The normal distributions transform: a new approach to laser scan matching,” in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2743–2748 vol.3, 2003.
- [17] E. Bonnal, *3D Mapping of Indoor Environments Using RGB-D Kinect Camera for Robotic Mobile Application*. Universitat Politècnica de Catalunya. Escola Tècnica Superior d’Enginyeria Industrial de Barcelona. Escola Tècnica Superior d’Enginyeria Industrial de Barcelona. Mobilitat, 2010 (Enginyeria Industrial), 2010.
- [18] A. Gil, O. Reinoso, M. Ballesta, and M. Juliá, “Multi-robot visual {SLAM} using a rao-blackwellized particle filter,” *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 68 – 80, 2010.
- [19] A. Vasebi, S. Bathaee, and M. Partovibakhsh, “Predicting state of charge of lead-acid batteries for hybrid electric vehicles by extended kalman filter,” *Energy Conversion and Management*, vol. 49, no. 1, pp. 75 – 82, 2008.
- [20] F. Sun, X. Hu, Y.Z., and S. Li, “Adaptive unscented kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles,” *Energy*, vol. 36, no. 5, pp. 3531 – 3540, 2011.
- [21] L. Ray, “Nonlinear tire force estimation and road friction identification: Simulation and experiments,” *Automatica*, vol. 33, no. 10, pp. 1819 – 1833, 1997.
- [22] F. Gustafsson, “Slip-based tire-road friction estimation,” *Automatica*, vol. 33, no. 6, pp. 1087 – 1099, 1997.

- [23] L. Pau, “A controlled linearized kalman filter for economic forecasting and adaptive modelling,” *Automatica*, vol. 14, no. 2, pp. 119 – 128, 1978.
- [24] J. Yim, C. Park, J. Joo, and S. Jeong, “Extended kalman filter for wireless {LAN} based indoor positioning,” *Decision Support Systems*, vol. 45, no. 4, pp. 960 – 971, 2008. jce:titlejInformation Technology and Systems in the Internet-Eraj/ce:titlej.
- [25] J. Yim, S. Jeong, K. Gwon, and J. Joo, “Improvement of kalman filters for {WLAN} based indoor tracking,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 426 – 433, 2010.
- [26] V. John, E. Trucco, and S. Ivekovic, “Markerless human articulated tracking using hierarchical particle swarm optimisation,” *Image and Vision Computing*, vol. 28, no. 11, pp. 1530 – 1547, 2010.
- [27] J. Czyz, B. Ristic, and B. Macq, “A particle filter for joint detection and tracking of color objects,” *Image and Vision Computing*, vol. 25, no. 8, pp. 1271 – 1281, 2007.
- [28] A. Ahmad and P. Lima, “Multi-robot cooperative spherical-object tracking in 3d space based on particle filters,” *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1084 – 1093, 2013. jce:titlejSelected Papers from the 5th European Conference on Mobile Robots (ECMR 2011)i/ce:titlej.
- [29] Y. Huang and J. Oliver, “Integrated simulation, error assessment, and tool path correction for five-axis {NC} milling,” *Journal of Manufacturing Systems*, vol. 14, no. 5, pp. 331 – 344, 1995.
- [30] O. Ilushin, G. Elber, D. Halperin, R. Wein, and M. Kim, “Precise global collision detection in multi-axis nc-machining,” *Computer-Aided Design*, vol. 37, no. 9, pp. 909 – 920, 2005. jce:titlejCAD’ 04 Special Issue: Product Design, Integration and Manufacturingi/ce:titlej.

- [31] I. Prodan, S. Olaru, R. Bencatel, J. de Sousa, C. Stoica, and S. Niculescu, “Receding horizon flight control for trajectory tracking of autonomous aerial vehicles,” *Control Engineering Practice*, vol. 21, no. 10, pp. 1334 – 1349, 2013.
- [32] E. Camacho, D. Ramirez, D. Limon, D. M. noz de la Peña, and T. Alamo, “Model predictive control techniques for hybrid systems,” *Annual Reviews in Control*, vol. 34, no. 1, pp. 21 – 31, 2010.
- [33] A. Bemporad and D. de la Peña, “Multiobjective model predictive control,” *Automatica*, vol. 45, no. 12, pp. 2823 – 2830, 2009.
- [34] T. Kraus, H. Ferreau, E. Kayacan, H. Ramon, J. D. Baerdemaeker, M. Diehl, and W. Saeys, “Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles,” *Computers and Electronics in Agriculture*, vol. 98, no. 0, pp. 25 – 33, 2013.
- [35] A. Amditis, E. Bertolazzi, M. Bimpas, F. Biral, P. Bosetti, M. Lio, L. Danielsson, A. Gallione, H. Lind, A. Saroldi, and A. Sjögren, “A holistic approach to the integration of safety applications: The insafes subproject within the european framework programme 6 integrating project prevent,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 554–566, 2010.
- [36] C. Urmson, J. Anhalt, H. Bae, J. A. D. Bagnell, R. Baker , R. Bittner, T. Brown, M. Clark, M. Darms, D. Demitrish, F. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, J. Struble, A. T. Stentz, M. Taylor , W. R. L. Whittaker, Z. Wolkowicki, W. Zhang, and J. Ziglar, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*

- Special Issue on the 2007 DARPA Urban Challenge, Part I*, vol. 25, pp. 425–466, June 2008.
- [37] E. Bertolazzi, F. Biral, and M. D. Lio, “real-time motion planning for multibody systems,” *Multibody System Dynamics*, vol. 17, no. 2-3, pp. 119–139, 2007.
 - [38] X. Wang and X. Xu, “An interoperable solution for cloud manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 4, pp. 232 – 247, 2013.
 - [39] D. Wu, M. Greer, D. Rosen, and D. Schaefer, “Cloud manufacturing: Strategic vision and state-of-the-art,” *Journal of Manufacturing Systems*, no. 0, 2013.
 - [40] P. Stavropoulos, D. Chantzis, C. Doukas, A. Papacharalampopoulos, and G. Chrysosolouris, “Monitoring and control of manufacturing processes: A review,” *Procedia CIRP*, vol. 8, no. 0, pp. 421 – 425, 2013.
 - [41] A. Ulsoy and Y. Koren, “Control of machining process,” *Transaction of the ASME*, vol. 115.
 - [42] S. Kurada and C. Bradley, “A review of machine vision sensors for tool condition monitoring,” *Computers in Industry*, vol. 34, no. 1, pp. 55 – 72, 1997.
 - [43] E. Budak and L. Kops, “Improving productivity and part quality in milling of titanium based impellers by chatter suppression and force control,” *{CIRP} Annals - Manufacturing Technology*, vol. 49, no. 1, pp. 31 – 36, 2000.
 - [44] U. Zuperl, F. Cus, and M. Milfelner, “Fuzzy control strategy for an adaptive force control in end-milling,” *Journal of Materials Processing Technology*, vol. 164â165, no. 0, pp. 1472 – 1478, 2005. jce:titlejAMPT/AMME05 Part 2j/ce:titlej.

- [45] I. Tansel, B. Ozcelik, W. Bao, P. Chen, D. Rincon, S. Yang, and A. Yenilmez, "Selection of optimal cutting conditions by using gonns," *International Journal of Machine Tools Manufacture*, vol. 46, pp. 26–35, 2005.
- [46] S. Aykut, M. Demetgul, and I. Tansel, "Selection of optimum cutting condition of cobalt-based superalloy with gonns," *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 9-12, pp. 957–967, 2010.
- [47] N. Sugita, T. Nakano, Y. Nakajima, K. Fujiwara, N. Abe, T. Ozaki, M. Suzuki, and M. Mitsuishi, "Dynamic controlled milling process for bone machining," *Journal of Materials Processing Technology*, vol. 209, no. 17, pp. 5777 – 5784, 2009.
- [48] T. Kim and J. Kim, "Adaptive cutting force control for a machining center by using indirect cutting force measurements," *International Journal of Machine Tools and Manufacture*, vol. 36, no. 8, pp. 925 – 937, 1996.
- [49] N. Wang, X. Ze, C. Ai, and Y. Ji, "The adaptive numerical control system based on cutting force restriction," in *Intelligent Systems Design and Applications, 2006. ISDA 2006. Sixth International Conference on*, vol. 2, pp. 213–218, 2006.
- [50] W. Hao, Z. Hongtao, G. Qianjian, W. Xiushan, and Y. Jianguo, "Thermal error optimization modeling and real-time compensation on a {CNC} turning center," *Journal of Materials Processing Technology*, vol. 207, no. 1â3, pp. 172 – 179, 2008.
- [51] H. Yang and J. Ni, "Adaptive model estimation of machine-tool thermal errors based on recursive dynamic modeling strategy," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 1, pp. 1 – 11, 2005.
- [52] P. Bosetti and S. Bruschi, "Enhancing positioning accuracy of cnc machine tools

- by means of direct measurement of deformation,” *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 5-8, pp. 651–662, 2012.
- [53] Y. Altintas and E. Budak, “Analytical prediction of stability lobes in milling,” *{CIRP} Annals - Manufacturing Technology*, vol. 44, no. 1, pp. 357 – 362, 1995.
- [54] G. Quintana and J. Ciurana, “Chatter in machining processes: A review,” *International Journal of Machine Tools and Manufacture*, vol. 51, no. 5, pp. 363 – 376, 2011.
- [55] J. Wang and K. Lee, “Suppression of chatter vibration of a cnc machine centre - an example,” *Mechanical Systems and Signal Processing*, vol. 10, no. 5, pp. 551 – 560, 1996.
- [56] K. Liu and K. Rouch, “Optimal passive vibration control of cutting process stability in milling,” *Journal of Materials Processing Technology*, vol. 28, no. 1&2, pp. 285 – 294, 1991.
- [57] N. H. Kim, D. Won, and J. C. Ziegert, “Numerical analysis and parameter study of a mechanical damper for use in long slender endmills,” *International Journal of Machine Tools and Manufacture*, vol. 46, no. 5, pp. 500 – 507, 2006.
- [58] S. Semercigil and L. Chen, “Preliminary computations for chatter control in end milling,” *Journal of Sound and Vibrations*, vol. 249, no. 3, pp. 1 – 12, 2002.
- [59] A. Rashid and C. Nicolescu, “Design and implementation of tuned viscoelastic dampers for vibration control in milling,” *International Journal of Machine Tools and Manufacture*, vol. 48, no. 9, pp. 1036 – 1053, 2008.
- [60] M. Pedro and P. Pahud, “The frahm damper,” in *Vibration Mechanics*, pp. 156–171, Springer Netherlands, 1991.

- [61] K. Shirase and Y. Altıntaş, “Cutting force and dimensional surface error generation in peripheral milling with variable pitch helical end mills,” *International Journal of Machine Tools and Manufacture*, vol. 36, no. 5, pp. 567 – 584, 1996.
- [62] A. Yusoff, S. Turner, C. Taylor, and N. Sims, “The role of tool geometry in process damped milling,” *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 9-12, pp. 883–895, 2010.
- [63] N. Cau, M. Leonesio, E. Zanotti, P. Parenti, and G. Bianchi, “Integrated machine tool design,” in *2nd IPROMS International Researchers Symposium*, pp. 22–24, 2009.
- [64] J. Dohner, J. Lauffer, T. Hinnerichs, N. Shankar, M. Regelbrugge, C. Kwan, R. Xu, B. Winterbauer, and K. Bridger, “Mitigation of chatter instabilities in milling by active structural control,” *Journal of Sound and Vibration*, vol. 269, no. 1&2, pp. 197 – 211, 2004.
- [65] E. Abele, H. Hanselka, F. Haase, D. Schlote, and A. Schiffler, “Development and design of an active work piece holder driven by piezo actuators.,” *Production Engineering*, no. 4, pp. 437–442.
- [66] H. Moradi, M. Movahhedy, and G. Vossoughi, “Robust control strategy for suppression of regenerative chatter in turning,” *Journal of Manufacturing Processes*, vol. 11, no. 2, pp. 55 – 65, 2009.
- [67] H. Moradi, G. Vossoughi, M. Movahhedy, and H. Salarieh, “Suppression of non-linear regenerative chatter in milling process via robust optimal control,” *Journal of Process Control*, vol. 23, no. 5, pp. 631 – 648, 2013.
- [68] P. Potočník and I. Grabec, “Nonlinear model predictive control of a cutting process,” *Neurocomputing*, vol. 43, no. 1&4, pp. 107 – 126, 2002. jce:titleSelected engineering applications of neural networksj/ce:title.

- [69] T. H. T. Takemura, T. Kitamura, “Active suppression of chatter by programmed variation of spindle speed,” in *Ann CIRP*, vol. 23, p. 121â122, 1974.
- [70] J. Sexton, R. Milne, and B. Stone, “A stability analysis of single-point machining with varying spindle speed,” *Applied Mathematical Modelling*, vol. 1, no. 6, pp. 310 – 318, 1977.
- [71] F. Ismail and E. Kubica, “Active suppression of chatter in peripheral milling part 1. a statistical indicator to evaluate the spindle speed modulation method,” *The International Journal of Advanced Manufacturing Technology*, vol. 10, no. 5, pp. 299–310, 1995.
- [72] K. Hajikolaie, H. Moradi, G. Vossoughi, and M. Movahhedy, “Spindle speed variation and adaptive force regulation to suppress regenerative chatter in the turning process,” *Journal of Manufacturing Processes*, vol. 12, no. 2, pp. 106 – 115, 2010.
- [73] P. Bosetti, M. Leonesio, and P. Parenti, “On development of an optimal control system for real-time process optimization on milling machine tools,” in *8th CIRP Conference on Intelligent Computation in Manufacturing Engineering*, 2012.
- [74] S. Mekid, P. Pruschek, and J. Hernandez, “Beyond intelligent manufacturing: A new generation of flexible intelligent {NC} machines,” *Mechanism and Machine Theory*, vol. 44, no. 2, pp. 466 – 476, 2009.
- [75] G. Amitay, S. Malkin, and Y. Koren, “Adaptive control optimization of grinding,” *Journal of Engineering for Industry*, vol. 103, no. 1, pp. 103–109, 1981.
- [76] Y. Qin and S. Park, “Robust adaptive control of machining operations,” in *Mechatronics and Automation, 2005 IEEE International Conference*, vol. 2, pp. 975–979 Vol. 2, 2005.

- [77] G. H. Kruger, A. J. Shih, D. G. Hattingh, and T. I. van Niekerk, “Intelligent machine agent architecture for adaptive control optimization of manufacturing processes,” *Advanced Engineering Informatics*, vol. 25, no. 4, pp. 783 – 796, 2011. *Special Section: Advances and Challenges in Computing in Civil and Building Engineering*.
- [78] U. Zuperl, F. Cus, B. Mursec, and T. Ploj, “A hybrid analytical-neural network approach to the determination of optimal cutting conditions,” *Journal of Materials Processing Technology*, vol. 157â158, no. 0, pp. 82 – 90, 2004. *Achievements in Mechanical and Materials Engineering Conference*.
- [79] F. Ridwan and X. Xu, “Advanced {CNC} system with in-process feed-rate optimisation,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 3, pp. 12 – 20, 2013.
- [80] X. Xu, H. Wang, J. Mao, S. Newman, T. Kramer, F. Proctor, and J. Michaloski, “Step-compliant nc research: the search for intelligent cad/capp/cam/cnc integration,” *International Journal of Production Research*, vol. 43, no. 17, pp. 3703–3743, 2005.
- [81] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli, “Autonomous vehicles control in the vislab intercontinental autonomous challenge,” *Annual Reviews in Control*, vol. 36, no. 1, pp. 161 – 171, 2012.
- [82] J. Troutman, *Variational Calculus and Optimal Control: Optimization With Elementary Convexity*. Undergraduate Texts in Mathematics Series, Springer-Verlag, 1996.
- [83] E. Frazzoli, M. A. Dahleh, and E. Feron, “Real-time motion planning for ag-

- ile autonomous vehicles,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 116–129, 2000.
- [84] “Chapter 7: Synthesis of time-optimal and minimum-effort control of linear delay systems,” in *Stability and Time-Optimal Control of Hereditary Systems* (E. Chukwu, ed.), vol. 188 of *Mathematics in Science and Engineering*, pp. 207 – 302, Elsevier, 1992.
- [85] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, Springer, 2006.
- [86] L. Grippo, F. Lampariello, and S. Lucidi, “A nonmonotone line search technique for newton’s method,” *SIAM J. Numer. Anal.*, vol. 23, pp. 707–716, Aug. 1986.
- [87] A. Iglesias, J. Munoa, and J. Ciurana, “Optimisation of face milling operations with structural chatter using a stability model based process planning methodology,” *International Journal of Advanced Manufacturing Technology*, pp. 1–13, 2013.
- [88] S. Park, Y. Altintas, and M. Movahhedy, “Receptance coupling for end mills,” *International Journal of Machine Tools and Manufacture*, vol. 43, no. 9, pp. 889–896, 2003.
- [89] F. Fritsch and R. Carlson, “Monotone piecewise cubic interpolation,” *SIAM: SIAM Journal on Numerical Analysis*, vol. 17, pp. 238 – 246, 1988.
- [90] S. Merdol and Y. Altintas, “Virtual simulation and optimization of milling applications—part ii: Optimization and feedrate scheduling,” *Journal of Manufacturing Science and Engineering*, vol. 130, no. 5, 2008.
- [91] A. Otto and G. Radons, “Application of spindle speed variation for chatter sup-

- pression in turning,” *CIRP Journal of Manufacturing Science and Technology*, vol. 6, no. 2, pp. 102–109.
- [92] M. Born and V. Fock, “Beweis des adiabatenatzes,” *Zeitschrift für Physik*, vol. 51, no. 3-4, pp. 165–180, 1928.
- [93] Y. Altintas, G. Stepan, D. Merdol, and Z. Dombovari, “Chatter stability of milling in frequency and discrete time domain,” *CIRP Journal of Manufacturing Science and Technology*, vol. 1, no. 1, pp. 35–44, 2008.
- [94] G. Boothroyd, P. Dewhurst, and W. Knight, *Product Design for Manufacture and Assembly*. Manufacturing engineering and materials processing, CRC Press, 2011.
- [95] P. Bosetti and F. Biral, “Rapid development of a CNC software within manufacturing automation courses,” in *Proceedings of International Mechanical Engineering Congress and Exposition* (ASME, ed.), pp. 1–10, 2008.
- [96] Y. Altintas, *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge University Press, 2000.
- [97] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [98] R. Merwe, *Sigma-point kalman filters for probabilistic inference in dynamic state-space models*. PhD thesis, 2004. AAI3129163.
- [99] M. Henning and S. Vinoski, *Advanced CORBA Programming with C++*. Reading, Massachusetts: Addison-Wesley, 1999.
- [100] D. Grisby, *The omniORB version 4.1 User’s Guide*. Cambridge: ATT Laboratories, 2009.

- [101] *Common Object Request Broker Architecture: Core Specification*. Object Management Group, Inc., 2002.
- [102] C. McHale, *CORBA Explained Simply*. Ciaran McHale, 2007.
- [103] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence, Prentice Hall, 2010.
- [104] D. Montgomery, *Design and Analysis of Experiments*. Student solutions manual, John Wiley & Sons, 2008.
- [105] G. Kerschen, K. Worden, A. Vakakis, and J. Golinval, “Past, present and future of nonlinear system identification in structural dynamics,” *Mechanical Systems and Signal Processing*, vol. 20, no. 3, pp. 505 – 592, 2006.
- [106] R. Bishop and D. Johnson, *The Mechanics of Vibration*. Cambridge University Press, 2011.
- [107] M. Steinbuch, R. van de Molengraft, and A. van der Voort, “Experimental modelling and lpv control of a motion system,” in *American Control Conference, 2003. Proceedings of the 2003*, vol. 2, pp. 1374–1379, 2003.
- [108] E. Balmes, “Frequency domain identification of structural dynamics using the pole/residue parametrization,” in *International Modal Analysis Conference*, pp. 540 – 546, 1996.
- [109] P. van Overschee and B. D. Moor, *Subspace Identification of Linear Systems: Theory, Implementation, Applications*. Springer Publishing, 1996.
- [110] D. ADAMS, “Frequency domain arx model and multi-harmonic frf estimators for non-linear dynamic systems,” *Journal of Sound and Vibration*, vol. 250, no. 5, pp. 935 – 950, 2002.

- [111] B. Wahlberg, M. Annergren, and C. Rojas, “On optimal input signal design for identification of output error models,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 5118–5124, 2011.
- [112] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [113] S. Vijayakumar, A. D’souza, and S. Schaal, “Incremental online learning in high dimensions,” *Neural Comput.*, vol. 17, pp. 2602–2634, Dec. 2005.
- [114] C. Rasmussen, “Gaussian processes for machine learning,” MIT Press, 2006.
- [115] R. Andrews, J. Diederich, and A. Tickle, “Survey and critique of techniques for extracting rules from trained artificial neural networks,” *Knowledge-Based Systems*, vol. 8, no. 6, pp. 373 – 389, 1995. [Knowledge-based neural networks](#).
- [116] C. Rasmussen and H. Nickisch, *The GPML Toolbox version 3.2*. jan 2013.
- [117] I. Bediaga, J. M. noa, J. Hernández, and L. L. de Lacalle, “An automatic spindle speed selection strategy to obtain stability in high-speed milling,” *International Journal of Machine Tools and Manufacture*, vol. 49, no. 5, pp. 384 – 394, 2009.
- [118] E. Kuljanic, M. Sortino, and G. Totis, “Multisensor approaches for chatter detection in milling,” *Journal of Sound and Vibration*, vol. 312, no. 4â5, pp. 672 – 693, 2008.
- [119] P. Norton and D. Karczub, *Fundamentals of Noise and Vibration Analysis for Engineers*. Cambridge University Press, 2003.
- [120] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

- [121] S. Ong, M. Yuan, and A. Nee, “Augmented reality applications in manufacturing: a survey,” *International Journal of Production Research*, vol. 46, pp. 2707–2742, Feb. 2008.
- [122] K. Weinert, A. Zabel, E. Ungemach, and S. Odendahl, “Improved nc path validation and manipulation with augmented reality methods,” *Production Engineering*, vol. 2, no. 4, pp. 371–376, 2008.
- [123] J. Zhang, S. Ong, and A. Nee, “Development of an ar system achieving in situ machining simulation on a 3-axis cnc machine,” *Comput. Animat. Virtual Worlds*, vol. 21, no. 2, pp. 103–115, 2010.
- [124] J. Zhang, S. Ong, and A. Nee, “Design and development of an in situ machining simulation system using augmented reality technology,” *Procedia {CIRP}*, vol. 3, no. 0, pp. 185 – 190, 2012. *{CIRP} Conference on Manufacturing Systems 2012*.
- [125] A. Olwal, J. Gustafsson, and C. Lindfors, “Spatial augmented reality on industrial CNC-machines,” in *Proceedings of SPIE 2008 Electronic Imaging*, vol. 6804, Jan. 2008.
- [126] M. Magnusson, *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, Årebro University, dec 2009. Årebro Studies in Technology 36.
- [127] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” vol. 10, (Newton, MA, USA), pp. 145–155, Butterworth-Heinemann, Apr. 1992.
- [128] P. Besl and N. McKay, “A method for registration of 3-d shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 239–256, Feb. 1992.

- [129] N. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, “Registration of point cloud data from a geometric optimization perspective,” in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, (New York, NY, USA), pp. 22–31, ACM, 2004.
- [130] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” in *UIST'11*, pp. 559–568, 2011.
- [131] R. Held, A. Gupta, and B. Curless, “3d puppetry: A kinect-based interface for 3d animation,” 2012.
- [132] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald, “Robust tracking for real-time dense RGB-D mapping with kintinuous,” Tech. Rep. MIT-CSAIL-TR-2012-031, Computer Science and Artificial Intelligence Laboratory, MIT, Sept. 2012.
- [133] A. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, (Washington, DC, USA), IEEE Computer Society, 2003.
- [134] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, (Washington, DC, USA), pp. 1–10, IEEE Computer Society, 2007.
- [135] P. Biber and W. Strasser, “The normal distributions transform: a new approach to laser scan matching,” in *Intelligent Robots and Systems, 2003. (IROS 2003)*.

- Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2743–2748, oct. 2003.
- [136] A. Diosi and L. Kleeman, “Laser scan matching in polar coordinates with application to slam,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3317 – 3322, aug. 2005.
 - [137] C. Ulas and H. Temeltas, “3d multi-layered normal distribution transform for fast and long range scan matching,” *Journal of Intelligent Robotic Systems*, pp. 1–24, 2012.
 - [138] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A state of the art in structured light patterns for surface profilometry,” *Pattern Recogn.*, vol. 43, pp. 2666–2680, Aug. 2010.
 - [139] T. Pribanić, S. Mrvoš, and J. Salvi, “Efficient multiple phase shift patterns for dense 3d acquisition in structured light scanning,” *Image Vision Comput.*, vol. 28, pp. 1255–1266, Aug. 2010.
 - [140] W. Li, S. Fang, and S. Duan, “3d shape measurement based on structured light projection applying polynomial interpolation technique,” *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 1, pp. 20 – 27, 2013.
 - [141] J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory, “Real-time 3d shape inspection system of automotive parts based on structured light pattern,” *Optics Laser Technology*, vol. 43, no. 1, pp. 1 – 8, 2011.
 - [142] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, P. K. A.J. Davison, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR '11*, (Washington, DC, USA), pp. 127–136, IEEE Computer Society, 2011.

- [143] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald, “Robust real-time visual odometry for dense RGB-D mapping,” in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, (Karlsruhe, Germany), Springer, May 2013. To appear.
- [144] Z. Kosmadoudi, T. Lim, J. Ritchie, S. Louchart, Y. Liu, and R. Sung, “Engineering design using game-enhanced cad: The potential to augment the user experience with game elements,” *Computer-Aided Design*, no. 0, pp. –, 2012.
- [145] A. Clapes, M. Reyes, and S. Escalera, “Multi-modal user identification and object recognition surveillance system,” *Pattern Recognition Letters*, no. 0, pp. –, 2012.
- [146] H. Chi, S. Kang, and X. Wang, “Research trends and opportunities of augmented reality applications in architecture, engineering, and construction,” *Automation in Construction*, no. 0, pp. –, 2013.
- [147] “Nvidia’s next generation cuda compute architecture: Kepler gk110,” tech. rep., NVidia Corporation, 2012.
- [148] H. Roth and M. Vona, “Moving volume kinectfusion,” in *Proceedings of the British Machine Vision Conference*, pp. 112.1–112.11, BMVA Press, 2012.
- [149] N. Govindaraju, S. Redon, M. Lin, and D. Manocha, “Cullide: interactive collision detection between complex models in large environments using graphics hardware,” in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, HWWS ’03, (Aire-la-Ville, Switzerland, Switzerland), pp. 25–32, Eurographics Association, 2003.
- [150] S. Gottschalk, M. L. D., and Manocha, “Obbtrees: a hierarchical structure for rapid interference detection,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’96, (New York, NY, USA), pp. 171–180, ACM, 1996.

- [151] J. Wang, D. Fleet, and A. Hertzmann, “Gaussian process dynamical models.”
- [152] B. Ferris, D. Fox, and N. Lawrence, “Wifi-slam using gaussian process latent variable models,” in *Proceedings of the 20th international joint conference on Artificial intelligence*, IJCAI’07, (San Francisco, CA, USA), pp. 2480–2485, Morgan Kaufmann Publishers Inc., 2007.
- [153] M. Salzmann and R. Urtasun, “Implicitly constrained gaussian process regression for monocular non-rigid pose estimation.”
- [154] K. Kim, D. Lee, and I. Essa, “Gaussian process regression flow for analysis of motion trajectories,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society, November 2011.
- [155] R. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1–4, 2011.
- [156] E. Budak, E. Armarego, and Y. Altintas, “Prediction of milling force coefficients from orthogonal cutting data,” *Journal of Engineering for Industry*, vol. 118, no. 2, pp. 216 – 224, 1996.